Numerical methods for electronic structure calculations

Dissertation for the degree of Doctor Philosophiae

Sergio Alberto Losilla Fernández
University of Helsinki
Faculty of Science
Department of Chemistry
Laboratory for Instruction in Swedish
P.O. Box 55 (A.I. Virtanens plats 1)
FI-00014 University of Helsinki, Finland

To be presented, with the assent of the Faculty of Science, University of Helsinki, for public discussion in Auditorium A129, Department of Chemistry (A.I. Virtanens plats 1, Helsinki), November the 15th, 2013, at noon.

Helsinki 2013

Supervised by

Prof. Dage Sundholm
Department of Chemistry
University of Helsinki
Helsinki, Finland

Reviewed by

Ove Christiansen
Department of Chemistry
Aarhus University
Aarhus, Denmark

Ville Havu
Department of Applied Physics
Aalto University School of Science
Espoo, Finland

Opponent

Jeppe Olsen
Department of Chemistry
Aarhus University
Aarhus, Denmark

ISBN 978-952-10-9437-8 (paperback)
ISBN 978-952-10-9438-5 (PDF)
http://ethesis.helsinki.fi
Unigrafia Helsinki 2013

I'm forever blowing bubbles,
Pretty bubbles in the air,
They fly so high,
Nearly reach the sky,
Then like my dreams,
They fade and die.

Fortune's always hiding,
I've looked everywhere,
I'm forever blowing bubbles,
Pretty bubbles in the air.

— James Kendis, James Brockman and Nat Vincent

Abstract

In this thesis, several numerical methods for electronic structure calculations are presented. The first is a quadrature scheme for the accurate and efficient computation of electrostatic potentials. The quadrature is applied to calculations on real-space grids, and to Coulomb integrals over Gaussian-type orbitals. Second, we introduce a real-space representation for three-dimensional scalar functions encountered in electronic structure calculations. In this representation, functions are partitioned into numerical atom-centred parts and a remainder, which is represented on a three-dimensional Cartesian grid. The algorithms to carry out the required operations are discussed, along with benchmarks of their computer implementations. The presented methods are all of a divide-and-conquer nature, breaking the problem into simple pieces which are suitable for execution in emerging massively parallel computer architectures, such as general-purpose graphics processing units.

List of Publications

List of original publications included in the thesis

I. Losilla, S. A.; Sundholm, D.; Jusélius, J. "The Direct Approach to Gravitation and Electrostatics Method for Periodic Systems", *J. Chem. Phys.* **132**, 024102 (2010).

II. Losilla, S. A.; Sundholm, D. "A divide and conquer real-space approach for all-electron molecular electrostatic potentials and interaction energies", *J. Chem. Phys.* **136**, 214104 (2012).

III. Losilla, S. A.; Mehine, M. M.; Sundholm, D. "Construction of the two-electron contribution to the Fock matrix by numerical integration", *Mol. Phys.* **110**, 2569 (2012).

IV. Mehine, M. M.; Losilla, S. A.; Sundholm, D. "An efficient algorithm to calculate three-electron integrals for Gaussian-type orbitals using numerical integration", *Mol. Phys.* **111**, 2536 (2013).

V. Losilla, S. A.; Watson, M. A.; Aspuru–Guzik, A; Sundholm, D. "GPGPU-accelerated real-space methods for molecular electronic structure calculations". Submitted to *J. Chem. Phys.*

Acknowledgements

In the first place, I would like to thank the people who have greatly helped to get this book into its current shape: Dage Sundholm, who endured the earliest version of the manuscript; my pre-reviewers Ove Christiansen and Ville Havu, both of whom gave insightful comments and corrected some embarrassing mistakes; Susi Lehtola, whose keen eye and sharp proofreading skills proved invaluable; Pekka Pyykkö, who undertook the task of pruning the most protruding branches of my gratuitously convoluted prose – and succeeded to a limited extent; and Sandra Söderholm, who has many reasons to be thanked for, but still spotted a few typos at the final stage.

I would like to deeply thank my supervisor, Dage Sundholm. This work would have been impossible without him. Not only because it is based on his work and ideas, and he has attentively supervised my research and helped me with any scientific problem I would have. Without his generosity and understanding, and without his kind words of support in moments of difficulty ("there are no problems, only solutions..."), this work, and these years, would not have been the same.

Jonas Jusélius, through the distance in both space in time, has had a larger influence on me than what he suspects. His first implementation of the DAGE program was a sort of Rosetta stone which opened for me the gates of how scientific software development should be done. He makes some wicked chicken wings, too.

In the last years, I have had the wonderful opportunity to supervise the research work of Mooses Mehine, Dou Du and Elias Toivanen. I hope I have been able to teach them one or two things, but I would like to thank them for all that they have taught me, for their friendship and the many good moments we have shared in the jätkähuone, and our continuing (I hope!) research collaboration.

Svenska Kemen has been an amazing environment to work in, despite its confusing name. This is thanks to its members, past and present, many of whom I consider good friends (in approximate order of appearance): Michael Patzschke, Patryk Zaleski-Ejgierd, Cong Wang, Ying-Chan Lin, Henrik Konschin, Sanna Lundberg, Sebastian Riedel, Mikael Johansson, Olli Lehtonen, Ville Kaila, Tommy Vänskä, Bertel Westermark, Anneka Tuomola, Nina Siegfrids, Michiko Atsumi, Raija Eskelinen, Raúl Mera, Heike Fliegl, Matej Pipíška, Nergiz Özcan, Stefan Taubert, Calle Suomivuori, Janne Pesonen, Krister Henriksson, Vasily Ochinnikov, Sussi Wiedmer, Annika Railila and Jana Lokajova, besides some people mentioned earlier. And of course, Teemu, Markus, Elinä and Richard, and the rest of the Kytketty Ryväs people.

One on the things that I have enjoyed the most during my PhD studies, largely thanks to the schools, conferences and research visits to which Dage encouraged me to go, is the great people that have crossed my path. My co-authors Alán Aspuru-Guzik and Mark Watson, the members of the research groups in Vienna, Harvard, Oslo and Tromsø, and others. The Magnus Ehrnrooths foundation is sincerely acknowledged for making this possible, by supporting many of these trips.

I feel very grateful to a number of people who have helped very much in sustaining my sanity during the long and dark Finnish winters: Marcelo, Olaya, Joonas and all the chemists, Kenty and Mónica, Sergio and Esther, Jokke and Maria and the bunch of spektrumiter, the members of the Chorus Sanctae Cecilia and Minna Nyberg. But of course, this would mean nothing, if it were not for those who, despite living far away, I feel very close: my grandmother Julita, my aunt Luisa, all my relatives, Mari Carmen and José Luis

and family, Gonzalo, Antonio, Víctor, Pablo, Marcos, Ximena, Carol, Sandra, Héctor, Covadonga, Cristina and others that I am forgetting and will remind me in a hopefully gentle manner. I want to thank my parents and my sister for always supporting me since I was a child. I would not be where I am nor what I am, if it were not for them.

And finally, thanks to you, Sandra. You have walked with me for almost as long as this thesis has taken to be completed. There is so much that you have brought into my life: a language, a culture, your parents Gunnel and Sture, Storlandet. Thank you for your love, for your support, for your great humour, for being as lovely as only you can be, for taking care of me: thank you for making me happy.

Helsinki, 27th October 2013

Contents

1 Introduction

The universe in numbers

Et ignem regunt numeri.

— Plato (attributed by Jean Baptiste Joseph Fourier, as cited in Théorie Analytique de la Chaleur)

Since the early times of geometry and arithmetics, humankind has been fascinated on how simple, analytic mathematical expressions can describe the world around us. The predictive power contained in compact and elegant algebraic expressions, such as Newton's laws of dynamics, the laws of thermodynamics, or Maxwell's equations, just to name a few, have mesmerized generations of science students.

But the realm of pure and eternal forms described by mathematical formulas is not the same realm which we perceive with our senses. The need of practical applicability pressed already in ancient times for the development of approximations to those ideal values (see Fig. 1.1), and led to the birth of numerical analysis.

As physics developed, it became clear that predictive power could only be obtained when sufficiently complex models were considered. The large amount of parameters and the difficulty of the equations rendered analytical solutions impractical, or even impossible. The 17th and 18th centuries saw the blooming of numerical methods, often aimed at predicting the motion of astronomical objects.

Today, there is yet one more powerful reason to develop efficient and accurate numerical methods: the most powerful computing tool in our hands, the digital computer, is only able to work in terms of discrete numbers, ultimately ones and zeros. Numerical analysis is the link to translating the physical equations into the language of digital computers. Moreover, the collapse of Moore's law and the emergence of massively parallel computer architectures presses even more for the production of algorithms that can be broken down into tasks which can be performed simultaneously.

Quantum chemistry: towards an *in silico* laboratory

In conclusion, I would like to emphasize strongly my belief that the era of computing chemists, when hundreds if not thousands of chemists will go to the computing machine instead of the laboratory for increasingly many facets of chemical information, is already at hand.

— Robert Sanderson Mulliken. Nobel Lecture, 1966

Scientists have long hoped for an *ab-initio* theory of chemical phenomena. In other words, to have the ability to compute the structure, properties and reactions of aggregates of matter with the minimal amount of imposed parameters, in the same way as the trajectories of astronomical objects can be predicted using Newtonian mechanics. Without the knowledge of the building blocks of matter, and the laws that govern their motion, such a chemical theory was barred. The advent of the atomic theory and quantum mechanics tore down these then impassable walls. Nevertheless, the reductionist program of the physicists, to

Figure 1.1 Babylonian tablet, ca. 1800–1600 BC. The symbols along the diagonal represent, in sexagesimal, $1 + 24/60 + 51/60^2 + 10/60^3 = 1.41421296\ldots$, which is an approximation to $\sqrt{2}$, accurate to 6 decimal figures. Yale Babylonian Collection. Photography: Bill Casselman.

"eat chemistry with a spoon" in the words of Heitler,* encountered numerical difficulties that proved to be formidable [2].

The 20th century saw quantum chemistry evolve from a discipline providing qualitative rules to assist experimental chemists [1], to providing accurate and reliable predictions, sometimes even surpassing the precision provided by experiments. The reason for this is twofold: on one hand, more and more accurate and efficient computational methods have made appearance; on the other, the gigantic leap in the power of computers has transformed calculations that were once inconceivably expensive into routine tasks.

Despite the advances, the full potential of quantum chemistry is yet to be unleashed: that of the *in silico* laboratory, being able to predict with experimental accuracy the properties of molecules and materials, and the reactions between them. The main obstacle is that the computational cost of tackling those systems with a sufficient precision is prohibitively expensive. The recent years have seen a spectacular success in lowering the scaling of the computational cost, and in adapting the available methods for the new generations of parallel machines. The quest for affordable, accurate computational methods is still ongoing.

*Letter from Heitler to London, cited in [1].

2 Electronic structure theory

The aim of this thesis is not to directly solve quantum chemical problems, but to provide a numerical framework for implementing practical solutions to them. In this section, we present the electronic structure calculation methods which are most commonly used to model molecular systems. A general and accessible review can be found in [3].

2.1 The electronic Schrödinger equation for many-atom systems

The time-independent Schrödinger equation describes the behaviour of non-relativistic physical systems in potentials that do not change with time:

$$\hat{H}\Psi = E\Psi. \tag{2.1}$$

$\hat{H}$ is the *Hamiltonian operator* of the system. Its eigenfunction Ψ, the *wave function*, depends on the degrees of freedom of the system. For an ensemble of particles with spin in three dimensions, $\Psi = \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, where $\mathbf{x}_i$ denotes both the position $\mathbf{r}_i = (x_i, y_i, z_i)$ and spin of the i-th particle. In general, there are infinitely many eigenfunctions Ψ, each corresponding to a possible state of the system. Being solutions of the time-independent Schrödinger equation, they are referred to as the *stationary states* of the system. The wave function is the coveted philosopher's stone of quantum mechanics: it encapsulates all the information that can be possibly extracted from the physical system. For each physical observable A, there is an associated mathematical operator $\hat{A}$, such that the expected value of a measurement of that observable, $\langle A \rangle$, is obtained as*

$$\langle A \rangle = \frac{\langle \Psi | \hat{A} | \Psi \rangle}{\langle \Psi | \Psi \rangle}. \tag{2.2}$$

For a normalized wave function, $\langle \Psi | \Psi \rangle = 1$. The Hamiltonian operator is associated with the total energy of the system, and hence the E appearing in (2.1) is the total energy of state Ψ. The eigenfunction with the lowest eigenvalue, or in other words, the state with the lowest energy E_0, Ψ_0, is referred to as the *ground state* of the system.

The obvious quantum mechanical approach to chemical problems, akin to solving Newton's equations of motion for a classical system, is to compute the wave function for a molecular system. For a molecule consisting of N electrons and K point-like nuclei, the Hamiltonian, in atomic units,† is given by

*(2.2) is only valid in general for time-dependent wave functions, or when $\hat{A}$ is time-independent. Some properties whose expectation values can be computed in this way are the linear momentum or the total energy of the system. Examples of properties which cannot be computed as expectation values of the time-independent wave function are response properties such as (hyper)polarizabilities. An approach to computing response properties from ground state wave functions is response-function theory [4].

†$e = 1$; $m_e = 1$; $\hbar = 1$; $1/(4\pi\epsilon_0) = 1$. The derived units $a_0 = 4\pi\varepsilon_0\hbar^2 m_e^{-1} e^{-2}$ and $E_h = m_e e^4 (4\pi\epsilon_0\hbar)^{-2}$ are also 1.

$$\hat{H} = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_A \frac{1}{2m_A}\nabla_A^2 + \sum_{i>j} \frac{1}{|\mathbf{r}_j - \mathbf{r}_i|} - \sum_{A,i} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|} + \sum_{A>B} \frac{Z_A Z_B}{|\mathbf{R}_B - \mathbf{R}_A|}, \quad (2.3)$$

where the indices i and j run over electrons, and A and B over nuclei.

For more than two particles, (2.1) cannot be solved analytically. As a first approximation, we can assume that the electrons perceive the nuclei, much heavier than electrons and hence much slower, as frozen in space. This assumption, known as the Born–Oppenheimer approximation [5], allows the separation of the Hamiltonian and the wave function into nuclear and electronic parts,

$$\hat{H} = \hat{H}_{\text{nuc}} + \hat{H}_{\text{el}} \quad (2.4)$$

and

$$\Psi = \psi_{\text{nuc}}\psi_{\text{el}}. \quad (2.5)$$

The electronic Hamiltonian contains the terms corresponding to the electronic kinetic energy, electron-electron repulsion and electron-nuclear attraction:

$$\hat{H}_{\text{el}} = -\frac{1}{2}\sum_i \nabla_i^2 + \sum_{i>j} \frac{1}{|\mathbf{r}_j - \mathbf{r}_i|} - \sum_{A,i} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|}. \quad (2.6)$$

A given molecular electronic Hamiltonian is thus completely defined by the number of electrons, and the positions and atomic numbers of the atomic nuclei. The electronic wave function ψ_{el} is the eigenfunction of the electronic Hamiltonian $\hat{H}_{\text{el}}$. We are left to solve the Schrödinger equation for the electrons:

$$\hat{H}_{\text{el}}\psi_{\text{el}} = E_{\text{el}}\psi_{\text{el}}. \quad (2.7)$$

We will concern ourselves exclusively with the calculation of electronic wave functions, and thus the el subscript will henceforth be dropped.

Electrons are fermions, that is, they are particles with half-integer spin, $S = 1/2$ in this case. This imposes the condition that the electronic wave function must be antisymmetric with respect to particle exchange. E.g., for swapping electrons 1 and 2,

$$\psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = -\psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_N). \quad (2.8)$$

This is known as Pauli's exclusion principle, as one of its most important consequences is that a quantum state can be occupied by only one fermion.

For many systems of interest, the first excited state lies a few eV (1 eV $\approx$ 27.1 E_h) above the ground state. Hence, most chemical phenomena can be explained in terms of the properties of the electronic ground state. For this purpose, we can exploit the variational principle, which establishes that for a given Hamiltonian $\hat{H}$ with ground state Ψ_0, and an arbitrary trial wave function $\tilde{\Psi}$

$$\frac{\langle\tilde{\Psi}|\hat{H}|\tilde{\Psi}\rangle}{\langle\tilde{\Psi}|\tilde{\Psi}\rangle} \geq \frac{\langle\Psi_0|\hat{H}|\Psi_0\rangle}{\langle\Psi_0|\Psi_0\rangle}. \quad (2.9)$$

The problem of finding the ground state wave function can then be regarded as a minimization problem. In other words, when the wave function is described in terms of a set of parameters, whose optimal values are those that provide a minimal energy.

2.2 The curse of dimensionality

Many computational problems found throughout all sorts of scientific disciplines are affected by the same type of blight: the computational cost grows exponentially with the number of dimensions. Most often, this increase of the workload is due to the amount of data that must be analysed. An illustrative example is the uniform sampling of a hypercube of side l in D dimensions. The amount of sampling points will be proportional to the volume, l^D. This problem is similar to constructing all possible sentences with D words with a vocabulary consisting of l entries. Even for very small vocabularies, as the number of words in the sentence grows, the amount of possible combinations becomes enormous. This phenomenon has been dubbed *the curse of dimensionality*.

Electronic structure calculations are indeed afflicted by this phenomenon. For a system consisting of N electrons, the electronic wave function, ψ, is a $3N$-dimensional scalar function. Consider a uniform sampling, where the function is represented on a $3N$-dimensional Cartesian grid using M points per side. The total amount of points is M^{3N}. For a system containing 10 electrons (such as H_2O), using a coarse grid of 100 points per dimension, just storing the ground state wave function requires 10^{60} coefficients. This exceeds by many orders of magnitude the world's total storage capacity in 2007, estimated to be about $3 \cdot 10^{20}$ bytes [6]. Of course, many systems of chemical interest have hundreds or thousands of electrons, and finer grids would be needed to provide a reliable description, rendering any such general approach unattainable.

In order to reduce the dimensionality of the problem, we write the wave function in terms of products of functions of lower dimension. Typically, these are one-particle functions, which depend on the coordinates of a single electron:

$$\psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \sum_m c_m \phi_1^m(\mathbf{x}_1) \phi_2^m(\mathbf{x}_2) \ldots \phi_N^m(\mathbf{x}_N). \tag{2.10}$$

The most common quantum chemical methods, which will occupy our attention for the remaining sections of this chapter, are based on (antisymmetrized) one-particle expansions like (2.10).

Despite the success of one-particle expansions, chemical accuracy (the error in energy typical of laboratory experiments, usually defined as 1 kcal/mol) requires a very large amount of terms in the expansion. As it was shown as early as in 1928 by Hylleraas [7, 8], including terms which depend on the coordinates of two electrons greatly reduces the number of terms needed to obtain a very high accuracy. Hylleraas proved that for the He atom (formed by a nucleus with charge $Z = 2$ and two electrons), the ground state can be faithfully represented by an expansion of the type

$$\psi(\mathbf{r}_1, \mathbf{r}_2) = \sum_m c_m \phi_1^m(\mathbf{r}_1) \phi_2^m(\mathbf{r}_2) \gamma_{12}^m(\mathbf{r}_1, \mathbf{r}_2). \tag{2.11}$$

The total ground state energy is reproduced to mE_h (ca. 0.6 kcal mol^{-1}) accuracy with only three terms [9]. The Hylleraas expansion can be generalized to model more complex systems to yield very accurate results [10–12]. Unfortunately, these approaches are prohibitively expensive, except for atoms and ions containing just a few electrons. Other approaches, referred to as *explicitly correlated methods* [13, 14], use simpler forms for the $\gamma_{12}^m(\mathbf{r}_1, \mathbf{r}_2)$ factors, such as r_{12} or $\mathrm{e}^{-\sigma r_{12}}$, to yield faster-converging versions of the one-particle expansion methods, which are discussed later in this section.

2.3 Self-consistent field methods

The simplest wave function of the form of (2.10) that fulfils the antisymmetry requirement (2.8) is the Slater determinant [15]:

$$\psi(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_N)=|\phi_1\phi_2\ldots\phi_N\rangle=\frac{1}{\sqrt{N!}}\begin{vmatrix}\phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N)\\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \cdots & \phi_2(\mathbf{x}_N)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_N(\mathbf{x}_1) & \phi_N(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_N)\end{vmatrix}. \tag{2.12}$$

The orthonormal one-particle functions $\phi_i(\mathbf{x})$, the *spin-orbitals*, can be expressed as a product of a three-dimensional spatial part and a spin part. A Slater determinant can be also written in the notation of (2.10) as

$$|\phi_1\phi_2\ldots\phi_N\rangle=\sum_{m=1}^{N!}\frac{\varepsilon_{a_{i1}a_{i2}\ldots a_{iN}}}{\sqrt{N!}}\phi_{a_{i1}}(\mathbf{x}_1)\phi_{a_{i2}}(\mathbf{x}_2)\ldots\phi_{a_{iN}}(\mathbf{x}_N), \tag{2.13}$$

where the coefficients c_m are given in terms of the N-dimensional Levi–Civita symbol $\varepsilon_{ij\ldots N}$, and a_{mn} is the n-th element of the m-th permutation of the set $\{1,2,\ldots,N\}$.

The energy of a Slater determinant is given by

$$\begin{aligned}\langle\phi_1\phi_2\ldots\phi_N|\hat{H}|\phi_1\phi_2\ldots\phi_N\rangle&=\sum_{i=1}^{N}\langle i|\hat{h}|i\rangle+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\langle ij|r_{12}^{-1}|ij\rangle-\langle ij|r_{12}^{-1}|ji\rangle\right)\\&=\sum_{i=1}^{N}\langle i|\hat{h}|i\rangle+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\langle i|\hat{J}_j-\hat{K}_j|i\rangle,\end{aligned} \tag{2.14}$$

where the 1- and 2- electron integrals are

$$\langle i|\hat{o}|j\rangle=\int\phi_i^*(\mathbf{x})\,\hat{o}\,\phi_j(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{2.15}$$

and

$$\langle ij|\hat{o}|kl\rangle=\int\int\phi_i^*(\mathbf{x}_1)\phi_j^*(\mathbf{x}_2)\,\hat{o}\,\phi_k(\mathbf{x}_1)\phi_l(\mathbf{x}_2)\,\mathrm{d}\mathbf{x}_1\,\mathrm{d}\mathbf{x}_2. \tag{2.16}$$

The operators appearing in (2.14) are the one-electron operator $\hat{h}$, which consists of the electron kinetic energy and the electron-nuclear interaction:

$$\hat{h}\phi_i(\mathbf{x})=\left(-\frac{1}{2}\nabla^2-\sum_A\frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|}\right)\phi_i(\mathbf{x}) \tag{2.17}$$

and the Coulomb and exchange operators, defined as

$$\hat{J}_j\phi_i(\mathbf{x})=\langle j|r_{12}^{-1}|j\rangle\,\phi_i(\mathbf{x}) \tag{2.18}$$

and

$$\hat{K}_j\phi_i(\mathbf{x})=\langle j|r_{12}^{-1}|i\rangle\,\phi_j(\mathbf{x}), \tag{2.19}$$

respectively. r_{12}^{-1} is shorthand notation for

$$r_{12}^{-1}=\frac{1}{|\mathbf{r}_2-\mathbf{r}_1|}. \tag{2.20}$$

The set of orbitals that minimize the energy of the Slater determinant can be obtained by means of Lagrange multipliers under the constraint that the spin-orbitals remain orthogonal, $\langle i|j\rangle=\delta_{ij}$, which yields the Hartree–Fock equations:

$$\hat{F}\phi_i(\mathbf{x}) = \varepsilon_i \phi_i(\mathbf{x}), \tag{2.21}$$

where the Fock operator is

$$\hat{F} = \hat{h} + \sum_{i=1}^{N} \left(\hat{J}_i - \hat{K}_i \right). \tag{2.22}$$

Because the Fock operator depends on the set of occupied orbitals, the equations defined by (2.21) must be solved iteratively, until some convergence criterion is met. In other words, the field caused by the electronic distribution eventually yields that same electron distribution. For this reason, the Hartree–Fock method is said to be a self-consistent field (SCF) method.

The Hartree–Fock method typically provides a reasonable qualitative description of molecules close to their equilibrium structures, recovering 95–99% of the total energy of the system. Nevertheless, typical molecular systems have total energies of a few tens E_h per electron, which means that a relative error of 1% supposes typically an absolute error of tens or hundreds of kcal/mol per electron. This immediately excludes the Hartree–Fock to many chemical problems.

The most important defficiency of Hartree–Fock theory is its mean-field character: in this model, electrons do not interact instantaneously with each other, but only with the average field created by the other electrons. This is due to the ansatz imposed: when the wave function has the form of a Slater determinant, the probability of finding an electron in a certain region of space is independent of the location of the other electrons. In other words, the motion of the electrons is not correlated.

Density functional theory

Based on the theorems of Hohenberg and Kohn [16], Kohn and Sham proved that the exact ground state density of an N-electron system in an external potential $v_\mathrm{ext}(\mathbf{r})$ (typically the nuclear potential) is identical to the density $\rho(\mathbf{r})$ of a system formed by N non-interacting particles moving in an effective potential given by [17]

$$v_\mathrm{eff}(\mathbf{r}) = v_\mathrm{ext}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \mathrm{d}^3 r' + v_\mathrm{xc}(\mathbf{r}), \tag{2.23}$$

where the *exchange–correlation potential* $v_\mathrm{xc}(\mathbf{r})$ is given in terms of the *exchange–correlation energy functional* $E_\mathrm{xc}[\rho]$

$$v_\mathrm{xc} = \frac{\delta E_\mathrm{xc}[\rho]}{\delta \rho(\mathbf{r})}. \tag{2.24}$$

The exact wave function for such a non-interacting system is given exactly by a Slater determinant. The approach is indeed very similar to the Hartree–Fock method, the difference being that, instead of N electrons moving in the average electrostatic field caused by the rest of the electrons, N fictitious fermionic particles move in some effective potential $v_\mathrm{eff}(\mathbf{r})$, which accounts also for correlation and exchange. The Kohn–Sham equations can be solved in an identical manner as the Hartree–Fock equations, where the Fock operator is given by

$$\hat{F} = \hat{h} + \sum_{i=1}^{N} \hat{J}_i + v_\mathrm{xc}. \tag{2.25}$$

$v_\mathrm{xc}(\mathbf{r})$ is a non-linear function of $\rho(\mathbf{r})$, therefore these integrals require sophisticated integration techniques [18–20]. This methodology is density functional theory (DFT).

The exact form of the exchange–correlation energy functional is not known, and needs to be approximated based on empirical data, other high accuracy electronic structure calculations, and knowledge of the properties that the exact functional must fulfil. A plethora of density functionals can be found in the literature, with different costs, accuracies, and ranges of applicability [21].

The main shortcoming of DFT is that, in its current form, it is not possible to predict when a given functional will fail, requiring extensive and careful benchmarking in each particular case. In practice, DFT provides a fairly good accuracy (comparable to some of the methods discussed later in this section) at a cost similar to Hartree–Fock calculations. Indeed, DFT calculations have been the bread-and-butter of quantum chemistry during the last decade.

2.4 Post-Hartree–Fock methods

We denote the Slater determinant constructed with the lowest N spin orbitals as the Hartree–Fock reference:

$$\Phi_0(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = |\phi_1 \phi_2 \ldots \phi_N\rangle \tag{2.26}$$

We can construct excited determinants by moving electrons from the occupied orbitals in the Hartree–Fock reference into unoccupied orbitals. This can be expressed in terms of single excitation operators

$$\hat{e}_i^a \Phi_0 = \hat{e}_i^a \left|\phi_1 \phi_2 \ldots \phi_b \phi_i \phi_j \ldots \phi_N\right\rangle = \left|\phi_1 \phi_2 \ldots \phi_b \phi_j \ldots \phi_N \phi_a\right\rangle, \tag{2.27}$$

double excitation operators

$$\hat{e}_{ij}^{ab} \Phi_0 = \hat{e}_i^a \hat{e}_j^b \Phi_0 = \hat{e}_i^a \hat{e}_j^b \left|\phi_1 \phi_2 \ldots \phi_b \phi_i \phi_j \phi_k \ldots \phi_N\right\rangle = |\phi_1 \phi_2 \ldots \phi_b \phi_k \ldots \phi_N \phi_a \phi_b\rangle, \tag{2.28}$$

etc. The full, infinitely large set of all the eigenfunctions of the Fock operator form a complete space. Therefore, the wave function can be written *exactly* as a linear combination of all possible Slater determinants that can be constructed:

$$\psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \sum_{ij\ldots rs} A_{ij\ldots rs} \left|\phi_i \phi_j \ldots \phi_r \phi_s\right\rangle. \tag{2.29}$$

This is the *full configuration interaction* (CI) ansatz. The expectation value for a such a wave function is

$$\langle\psi|\hat{H}|\psi\rangle = \sum_{ij\ldots rs} \sum_{i'j'\ldots r's'} A_{ij\ldots rs} H_{ij\ldots rs}^{i'j'\ldots r's'} A_{i'j'\ldots r's'} \tag{2.30}$$

where the Hamiltonian matrix elements are

$$H_{ij\ldots rs}^{i'j'\ldots r's'} = \left\langle \phi_i \phi_j \ldots \phi_r \phi_s \right| \hat{H} \left| \phi_{i'} \phi_{j'} \ldots \phi_{r'} \phi_{s'} \right\rangle. \tag{2.31}$$

The CI coefficients $A_{ij\ldots rs}$ can be computed by solving the secular CI equations:

$$\sum_{i'j'\ldots r's'} H_{ij\ldots rs}^{i'j'\ldots r's'} A_{i'j'\ldots r's'} = E A_{ij\ldots rs}. \tag{2.32}$$

The dimensionality of the full configuration interaction tensor is asymptotically proportional to $N!$, or $\mathcal{O}(N!) \sim \mathcal{O}(e^N)$. The CI approach reintroduces the curse of dimensionality that was removed by representing the wave function as a Slater determinant. This renders the approach intractable except for the smallest systems. However, the vast majority of the elements of the full configuration interaction tensor are actually negligible. This

is because, for large N, most determinants represent high order excitations with respect to the Hartree–Fock reference state, and will contribute very little to the ground state wave function. Fortunately, there are approaches that allow including only the configurations that noticeably contribute to the description of the ground state, and neglect (or include in an approximate fashion) those which do not.

Generally speaking, the importance of the contributions, in terms of the size of the configuration interaction tensor element $|A_{ij\ldots rs}|$, decrease with the excitation order beyond doubles. We can reorder the terms in the configuration interaction wave function as

$$\begin{aligned}\psi(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N) = &\Phi_0 \\ &+\sum_{ia} C_i^a \hat{e}_i^a \Phi_0 \\ &+\sum_{ijab} C_{ij}^{ab} \hat{e}_{ij}^{ab} \Phi_0 \\ &+\sum_{ijkabc} C_{ijk}^{abc} \hat{e}_{ijk}^{abc} \Phi_0 + \ldots\end{aligned} \tag{2.33}$$

The elements of the configuration interaction tensor are now reorganized into more compact single, double, etc. excitation tensors, e.g.:

$$A_{2\ldots Na} = C_1^a \tag{2.34}$$

$$A_{34\ldots Nab} = C_{12}^{ab} \tag{2.35}$$

$$A_{456\ldots Nabc} = C_{123}^{abc}. \tag{2.36}$$

By neglecting all excited determinants above a certain order, different truncated configuration interaction methods are obtained. For instance, in the CISD (*configuration interaction including single and double excitations*) method, the elements of triple and higher order excitations tensors are set to zero. Notice that truncating the configuration interaction expansion to include only single excitations yields the same ground state energy as Hartree–Fock itself. This is due to Brillouin's theorem, which states that singly excited determinants do not interact with the Hartree–Fock ground state. In other words,

$$\langle \Phi_0 | \hat{H} \hat{e}_i^a | \Phi_0 \rangle = 0. \tag{2.37}$$

Singly excited determinants are however crucial for a correct description of the wave function, as they can interact with higher order excited determinants.

An alternative and equivalent way of writing the full CI wave function in (2.29) is the exponential *coupled cluster* ansatz [22]:

$$\psi(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N) = \mathrm{e}^{\mathbf{T}} \Phi_0(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N) \tag{2.38}$$

where the cluster operator $\mathbf{T}$ is defined as:

$$\mathbf{T} = \sum_i \mathbf{T}_i = \sum_{ia} t_i^a \hat{e}_i^a + \sum_{ijab} t_{ij}^{ab} \hat{e}_{ij}^{ab} + \sum_{ijkabc} t_{ijk}^{abc} \hat{e}_{ijk}^{abc} + \ldots \tag{2.39}$$

The exponential $\mathrm{e}^{\mathbf{T}}$ is

$$\mathrm{e}^{\mathbf{T}} = \sum_{n=0}^{\infty} \frac{\mathbf{T}^n}{n!} = 1 + \mathbf{T} + \frac{\mathbf{T}^2}{2} + \frac{\mathbf{T}^3}{6} + \ldots \tag{2.40}$$

The coefficients $t_{i\ldots}^{a\cdots}$, known as the *cluster amplitudes*, can be determined from a set of nonlinear equations. Because the Hamiltonian contains only one and two-electron operators,

these equations contain the amplitudes to at most fourth order. The CI tensor can be then again expressed in terms of lower dimensionality tensors, for instance

$$A_{2\ldots Na} = t_1^a \tag{2.41}$$

$$A_{34\ldots Nab} = t_{12}^{ab} + \frac{1}{2}\left(t_1^a t_2^b + t_1^b t_2^a\right) \tag{2.42}$$

$$A_{456\ldots Nabc} = t_{123}^{abc} + \frac{1}{2}\left(t_1^a t_{23}^{bc} + t_1^b t_{23}^{ac} + t_1^c t_{23}^{ab} + \ldots\right) + \frac{1}{6}\left(t_1^a t_2^b t_3^c + t_1^a t_2^c t_3^b + t_1^b t_2^a t_3^c + \ldots\right). \tag{2.43}$$

Similarly as in configuration interaction, excitations can be truncated to a certain order. The crucial difference between truncated configuration interaction and coupled cluster approaches becomes clear by comparing (2.34)–(2.36) to (2.41)–(2.43): in truncated coupled cluster methods, the full CI tensor is implicitly constructed. For instance, even if the cluster operator includes only single and double excitations (which yields the CCSD method), triply excited determinants will enter the wave function, as seen from (2.43). One of the most important consequences is that, unlike truncated configuration interaction methods, truncated coupled cluster methods are *size consistent*. In other words, as the system grows, larger order excitations become more and more important, because the probability of e.g. multiple pairs of electrons being excited simultaneously increases. Because of this reason, coupled cluster methods usually provide better accuracy (more so as the size of the system increases) for a similar cost, as compared to truncated configuration interaction approaches.

The computational cost of these truncated schemes scales polynomially with the system size, $\mathcal{O}(N^a)$, with $a \geq 6$. This is clearly a major improvement with respect to the exponential growth of the cost of full configuration interaction. Nevertheless, the scaling is steep, limiting the applicability of such approaches to small and mid-sized molecules. Another limitation is that very high order excitations must be included for a correct description of systems where the Hartree–Fock reference is not dominant. For these systems, multi-reference methods, which will be discussed below, are a more sensible alternative.

Other methods

Perturbative methods

Corrections to a reference wave function can be computed in a non-iterative manner using perturbation theory. Møller–Plesset perturbation theory [23], and most representatively its second order form (MP2), is a common way to add correlation to a reference Hartree–Fock wave function. Another remarkable example is the CCSD(T) method [24], which provides corrections arising from triple excitations to the CCSD wave function. The CCSD(T) method has been dubbed *the golden standard of quantum chemistry*, as it is considered the state of the art in terms of accuracy and reliability for energies and structures.

Multi-reference methods

The methods described above perform well when the Hartree–Fock state is the dominant one or, in other words, when the system is well described by a single Slater determinant. However, in the presence of low-lying excited states that interact strongly with the ground state this assumption breaks down [3]. Such systems include those in far-from-equilibrium geometric configurations and near-degenerate ground states. In multi-reference methods, each of those states is represented by a determinant constructed with a corresponding set of orbitals. Some representative multi-reference methods are the complete active space self-consistent field method (CAS-SCF), multi-reference configuration interaction (MRCI) and multi-configurational second-order perturbation theory (CASPT2) [25].

3 Electrostatic potentials

Electrostatic phenomena have been known for many centuries: the classic example is how amber ($\acute{\eta}\lambda\varepsilon\kappa\tau\rho\omega\nu$, *elektron*, in Greek), when rubbed against wool, attracts small pieces of materials such as hair or paper. But it would not be until the late 18th and early 19th century when Charles-Augustin de Coulomb, Michael Faraday and others discovered the equations governing the interactions between charged particles. These efforts would culminate in the late 19th century, when James Clerk Maxwell explained –and predicted– a large variety of physical phenomena under the umbrella of his four laws of electromagnetism [26].

Ordinary matter consists of charged particles: positively charged nuclei and negatively charged electrons. In this chapter, electrostatic potentials will be introduced in the context of electronic structure theory. We will also present one of the main results of this thesis, first introduced in Publication I: a universal method for calculating electrostatic potentials by direct integration of the Coulomb potential. Finally, we will show how the method can be easily extended to deal with screened Coulomb potentials, i.e., to solve Helmholtz's equation.

3.1 Electrostatics

The force experienced by a point charge q located at a position $\mathbf{r}$ is given by

$$\mathbf{F} = -q\nabla V(\mathbf{r}) \tag{3.1}$$

where $V(\mathbf{r})$ is the electrostatic potential. The corresponding potential energy is given by

$$E = qV(\mathbf{r}). \tag{3.2}$$

The electrostatic potential caused by a continuous charge distribution is determined by Poisson's equation:

$$\nabla^2 V(\mathbf{r}) = -4\pi\rho(\mathbf{r}). \tag{3.3}$$

The solution of Poisson's equation for a unit point charge ($\rho(\mathbf{r}) = \delta^3(\mathbf{r}-\mathbf{r}')$), subject to the boundary condition $\lim_{|\mathbf{r}-\mathbf{r}'|\to\infty} V(\mathbf{r}) = 0$, can be computed analytically as:

$$V(\mathbf{r}) = \frac{1}{|\mathbf{r}-\mathbf{r}'|}. \tag{3.4}$$

Equation (3.4) is also referred to as the Coulomb potential. Using (3.1), we obtain Coulomb's law, which gives the force $\mathbf{F}_{ij}$ experienced by a point charge q_j at position $\mathbf{r}_j$ in the presence of another point charge q_i at position $\mathbf{r}_i$:

$$\mathbf{F}_{ij} = \frac{q_i q_j}{r_{ij}^2}\hat{\mathbf{r}}_{ij} \tag{3.5}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$, $r_{ij} = |\mathbf{r}_{ij}|$ and $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$.

The Coulomb potential is the Green's function for solving Poisson's equation:

$$V(\mathbf{r}) = \int_{\mathbb{R}^3} G(\mathbf{r};\mathbf{r}')\rho(\mathbf{r}')\mathrm{d}^3 r' = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}^3 r'. \tag{3.6}$$

The interaction energy between two charge densities, $\rho_1(\mathbf{r})$ and $\rho_2(\mathbf{r})$, can be likewise written as

$$E = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{\rho_1(\mathbf{r}')\rho_2(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}^3 r\,\mathrm{d}^3 r' = \int_{\mathbb{R}^3} \rho_1(\mathbf{r})V_2(\mathbf{r})\mathrm{d}^3 r = \int_{\mathbb{R}^3} V_1(\mathbf{r})\rho_2(\mathbf{r})\mathrm{d}^3 r. \tag{3.7}$$

As discussed in Chapter 2, matter is formed by positively charged nuclei and negatively charged electrons. Electrostatics thus plays a fundamental role in the electronic structure of molecules. Several of the quantities discussed earlier have classical analogues. For instance, the integrals of the type

$$\langle i|\, r_{12}^{-1}\,|j\rangle = \int_{\mathbb{R}^3} \frac{\phi_i^*(\mathbf{r}_1)\phi_j(\mathbf{r}_1)}{|\mathbf{r}_1-\mathbf{r}_2|}\mathrm{d}^3 r_1 \tag{3.8}$$

appearing in the expressions of the Coulomb and exchange operators (2.18) and (2.19) can be interpreted as the electrostatic potentials caused by the charge distribution given by $\phi_i^*(\mathbf{r}_1)\phi_j(\mathbf{r}_1)$. Likewise, the Hartree potential which appears in the DFT effective potential in (2.23), is the electrostatic potential created by the ensemble of electrons, with a factor of $1/2$ to avoid double-counting. Similarly, the integrals

$$\langle ij|\, r_{12}^{-1}\,|kl\rangle = \int_{\mathbb{R}^3} \phi_i^*(\mathbf{r}_1)\phi_k(\mathbf{r}_1)\int_{\mathbb{R}^3} \frac{\phi_j^*(\mathbf{r}_2)\phi_l(\mathbf{r}_2)}{|\mathbf{r}_1-\mathbf{r}_2|}\mathrm{d}^3 r_2\mathrm{d}^3 r_1 \tag{3.9}$$

are the electrostatic interaction energy between the charge distributions $\phi_i^*(\mathbf{r}_1)\phi_k(\mathbf{r}_1)$ and $\phi_j^*(\mathbf{r}_2)\phi_l(\mathbf{r}_2)$.

3.2 The Laplace expansion of the Coulomb potential

The r^{-1} operator can be rewritten in terms of spherical harmonics [27]:

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{l=0}^{\infty}\frac{4\pi}{2l+1}\sum_{m=-l}^{l} Y_{lm}^*(\theta,\phi)Y_{lm}(\theta',\phi')\frac{r_<^l}{r_>^{l+1}}. \tag{3.10}$$

The spherical coordinates r, θ and ϕ are relative to the arbitrarily chosen origin. $r_<$ and $r_>$ are, respectively, the smaller and larger of $|\mathbf{r}|$ and $|\mathbf{r}'|$. The expression in (3.10) is referred to as the Laplace expansion of the Coulomb potential. The angular functions $Y_{lm}(\theta,\phi)$ are the spherical harmonics.

In molecular electronic structure calculations, it is common to encounter charge densities which can be separated in the form

$$\rho(\mathbf{r}) = \sum_{lm}\rho^{lm}(r)Y_{lm}(\theta,\phi). \tag{3.11}$$

Using the orthogonality property of the spherical harmonics

$$\langle Y_{lm}|Y_{l'm'}\rangle = \frac{2l+1}{4\pi}\delta_{ll'}\delta_{mm'} \tag{3.12}$$

and inserting (3.10) and (3.11) into (3.6), the electrostatic potential is obtained also in a form separated into radial and angular parts,

$$V(\mathbf{r}) = \sum_{lm} V^{lm}(r)Y_{lm}(\theta,\phi), \tag{3.13}$$

where the radial function $V(r)$ is given by

$$V^{lm}(r) = \frac{4\pi}{2l+1}\left[r^{-(l+1)}\int_0^r \rho^{lm}(s)s^{l+2}\mathrm{d}s + r^l\int_r^\infty \rho^{lm}(s)s^{1-l}\mathrm{d}s\right]. \tag{3.14}$$

The multipole expansion

Using (3.10), an electrostatic potential can be computed in the form of a *multipole expansion*:

$$V(\mathbf{r}) \approx \sum_{l=0}^{\infty}\frac{1}{r^{l+1}}\sum_{m=-l}^{l} q_{lm}Y_{lm}(\theta,\phi) \tag{3.15}$$

where the *multipole moments* q_{lm} are defined as

$$q_{lm} = \int_{\mathbb{R}^3}\rho(\mathbf{r})Y_{lm}(\theta,\phi)r^l\mathrm{d}^3r. \tag{3.16}$$

If the density is completely contained inside a sphere of radius R, i.e. $\rho(\mathbf{r}) = 0\ \forall\ \mathbf{r} : |\mathbf{r}| \geq R$, the multipole expansion (3.15) holds exactly outside that sphere.

The advantage of the multipole expansion can be seen as data compression: the structure of $\rho(\mathbf{r})$ is encoded into a mere collection of $(L+1)^2$ coefficients and the coordinates of the expansion center, where L is the truncation value for l. By means of the spherical harmonics addition theorem, it is also possible to combine several multipole expansions with different centers into one [28]. The interaction between two charge densities contained in non-overlapping spheres can be efficiently computed with an expression of the form [28]:

$$E = \sum_{lm}\sum_{l'm'} q_{lm}T_{lm}^{l'm'}q'_{l'm'}. \tag{3.17}$$

where $T_{lm}^{l'm'}$ denote the elements of the so-called interaction matrix.

Multipole expansions are the basis for the well-known fast multipole method [29]. Computing the electrostatic interaction between N charge densities would naively cost $\mathcal{O}(N^2)$ operations. The fast multipole method reduces the cost to $\mathcal{O}(N)$. Owing to this, the multipole expansion has been successfully applied in quantum chemical calculations on large systems [30–32].

3.3 Numerical calculation of electrostatic potentials

The calculation of electrostatic potentials is ubiquitous in the modelling of matter. It is not surprising that a variety of different approaches for the efficient calculation of electrostatic potentials have been developed over the years [33–43]. Likewise, there has been a lot of effort on the related problem of solving the Poisson–Boltzmann equation to compute the electrostatic potential caused by a charge distribution embedded in a ionic solution, of great importance in biological simulations [44–46].

Equations (3.3) and (3.6) offer two alternative roads for computing electrostatic potentials from a known charge density: either solving Poisson's equation –a linear partial differential equation– or directly integrating the Coulomb potential.

The direct integration of the Coulomb potential is, at first sight, the most simple and straightforward approach. For many analytic charge distributions, analytic integration can be carried out in a more or less straightforward manner. The multipole expansion of the Coulomb potential, discussed below, greatly simplifies the calculation of spherically symmetric charge distributions. But for general charge distributions, the integral is not solvable. We must therefore resort to some numerical integration scheme.

Let us consider discretizations of the type

$$\rho(\mathbf{r}) \approx \sum_i \rho(\mathbf{r}_i)\delta^3(\mathbf{r}-\mathbf{r}_i) \tag{3.18}$$

in which the charge density $\rho(\mathbf{r})$ is approximated as a collection of point charges located at N selected positions $\{\mathbf{r}_i\}$, the *grid points*.

The most straightforward approach is the direct summation of the Coulomb potential, that is

$$V(\mathbf{r}_i) \approx \sum_{i'} \frac{\rho(\mathbf{r}_i)}{|\mathbf{r}_i - \mathbf{r}_{i'}|}. \tag{3.19}$$

This approach is beset by two major obstacles. The first one is of numerical nature: how to deal with the singularity at $\mathbf{r}_i = \mathbf{r}_{i'}$. Typically, some numerical approximation is required to circumvent this problem [33]. The second issue is a matter of computational cost: to directly compute the potential at point $\mathbf{r}_i$ involves the calculation of N^2 contributions. An approach with a cost that scaled linearly with the size of the system would be clearly desirable.

For this reason, solutions of the Poisson equation (3.3) are commonly preferred. One popular and efficient way to solve it is the *finite difference* method. In this approach, the grid points are set on a Cartesian grid as $\mathbf{r}_{ijk} = (x_i, y_j, z_k)$. The Laplacian of a function is approximated as

$$\begin{aligned}&\nabla^2 f(x_i, y_j, z_k) \approx \\ &\sum_{n_x=-N}^{n} C_{n_x} f(x_{i+n_x}, y_j, z_k) + \sum_{n_y=-N}^{n} C_{n_y} f(x_i, y_{j+n_y}, z_k) + \sum_{n_z=-N}^{n} C_{n_z} f(x_i, y_j, z_{k+n_z})\end{aligned} \tag{3.20}$$

in terms of the real coefficients $\left\{C_{n_x}\right\}$, $\left\{C_{n_y}\right\}$ and $\left\{C_{n_z}\right\}$. Poisson's equation (3.3) can then be recast in a matrix form:

$$\mathbf{L}\mathbf{v} = -4\pi\mathbf{d}. \tag{3.21}$$

In (3.21), $\mathbf{d}$ and $\mathbf{v}$ contain the values of the density and potential at the grid points, respectively. The $\mathbf{L}$ matrix is constructed with the C_n coefficients in (3.20). The problem is therefore reduced to a system of linear equations. As for any differential equation, the boundary conditions have to be fixed in some way. Typically, this is done by explicitly computing the values of the potential at the grid boundaries via direct summation, multipole expansion, etc.

Poisson's equation can be solved by noting that the Fourier transform of the Laplacian of a function $f(\mathbf{r})$ is

$$\mathscr{F}\left(\nabla^2 f(\mathbf{r})\right) = -4\pi|\mathbf{k}|^2 \mathscr{F}(f(\mathbf{r})). \tag{3.22}$$

Therefore, the potential can be computed as

$$V(\mathbf{r}) = \mathscr{F}^{-1}\left(\frac{\mathscr{F}(\rho(\mathbf{r}))}{|\mathbf{k}|^2}\right). \tag{3.23}$$

Due to the availability of Fast Fourier Transform (FFT) algorithms to carry out very efficient direct and inverse Fourier transforms, with an $\mathscr{O}(N \log N)$ cost, (3.23) is a very attractive approach to compute electrostatic potentials. Unfortunately, there are certain shortcomings that make this approach unsuitable for all-electron molecular calculations. First, FFT algorithms require as input the values of the charge density in an evenly spaced Cartesian grid. The charge densities spanned by all the electrons in a molecule have sharp cusps, which would require points very close to each other. A Cartesian grid enveloping

the whole molecular electron density with such a small grid interval would contain a very large amount of points. Second, FFT algorithms assume that the charge density is periodic. A typical way to circumvent this issue and tackle isolated (non-periodic) systems using FFT-based Poisson solvers is encasing the density in a larger box to reduce the interference from the periodic images. This approach, however, cannot provide very accurate results, because the Coulomb potential decays slowly with the distance, and the elimination of the interferences would require huge boxes.

In the following, we will show that using numerical approximations, it is possible to compute electrostatic potentials in an accurate and efficient manner by directly integrating the Coulomb potential.

The Gaussian integral identity for the Coulomb potential

The singularity in the Coulomb operator can be circumvented by using the integral identity introduced by Singer [47]:

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \frac{2}{\sqrt{\pi}} \int_0^\infty \mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2} \mathrm{d}t. \tag{3.24}$$

Another convenient property of the integral identity, is that in this form the Coulomb operator is separable in the three Cartesian dimensions:

$$\mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2} = \mathrm{e}^{-t^2(x-x')^2} \mathrm{e}^{-t^2(y-y')^2} \mathrm{e}^{-t^2(z-z')^2}, \tag{3.25}$$

Eq. (3.24) has been extensively used by the electronic structure community, most notably to derive analytical [48–50] and numerical [51] schemes to compute Coulomb integrals involving Gaussian-type orbitals.

Inserting (3.24) into (3.6), we obtain the following expression for the electrostatic potential:

$$V(\mathbf{r}) = \frac{2}{\sqrt{\pi}} \int_0^\infty \int_{\mathbb{R}^3} \mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2} \rho(\mathbf{r}') \mathrm{d}^3 r' \mathrm{d}t \tag{3.26}$$

In this way, the direct integration of the singularity is avoided, at the cost of integrating in one additional dimension.

We now wish to devise an efficient and accurate strategy to evaluate the integral in t space. The first step is to analyse the properties of the integrand, which we shall denote as $W(\mathbf{r}, t)$:

$$W(\mathbf{r}, t) = \int_{\mathbb{R}^3} \mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2} \rho(\mathbf{r}') \mathrm{d}^3 r'. \tag{3.27}$$

Note that the value of t will determine how spread or tight the Gaussian operator is. Large values of t represent short range contributions to the electrostatic interaction, while small values of t will introduce contributions from the whole density, however far.

For sufficiently small t, we can approximate the Gaussian operator as a power series around $t = 0$:

$$\mathrm{e}^{-t^2 r^2} = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} (rt)^{2k} \approx 1 - (rt)^2 + \frac{(rt)^4}{2} - \frac{(rt)^6}{6} + \dots \tag{3.28}$$

Inserting (3.28) into (3.27), we obtain

$$W(\mathbf{r}, t) \approx \sum_{k=0}^{\infty} \frac{Q_k(\mathbf{r})(-1)^k}{k!} t^{2k} \approx Q_0(\mathbf{r}) - Q_1(\mathbf{r}) t^2 + \frac{Q_2(\mathbf{r}) t^4}{2} + \mathcal{O}(t^6) \tag{3.29}$$

where

$$Q_k(\mathbf{r}) = \int_{\mathbb{R}^3} (\mathbf{r} - \mathbf{r}')^{2k} \rho(\mathbf{r}') \mathrm{d}^3 r' \tag{3.30}$$

For t sufficiently small, $W(\mathbf{r}, t)$ is therefore an even polynomial in t. At the origin, $W(\mathbf{r}, 0)$ is equal to the total charge of the system $Q_0(\mathbf{r}) = Q = \int_{\mathbb{R}^3} \rho(\mathbf{r}') \mathrm{d}^3 r'$.

For very large t, we expand $\rho(\mathbf{r}')$ in a three-dimensional Taylor series. For a function $f(\mathbf{r}')$ expanded around point $\mathbf{r}' = \mathbf{r}$, the expansion is given by

$$f(\mathbf{r}') = \sum_{k=0}^{\infty} \sum_{\{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=k\}} \frac{\left[\partial_{\mathbf{r}'}^{\boldsymbol{\alpha}} f(\mathbf{r}')\right]_{\mathbf{r}'=\mathbf{r}}}{\boldsymbol{\alpha}!} (\mathbf{r}' - \mathbf{r})^{\boldsymbol{\alpha}}. \tag{3.31}$$

In (3.31) the multi-index $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3) \tag{3.32}$$

with $\{\alpha_1, \alpha_2, \alpha_3\} \in \mathbb{Z}^*$. The following notation is used:

$$|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \alpha_3 \tag{3.33}$$

$$n\boldsymbol{\alpha} = (n\alpha_1, n\alpha_2, n\alpha_3), \text{ for } n \in \mathbb{Z}^* \tag{3.34}$$

$$\boldsymbol{\alpha}! = \alpha_1! \alpha_2! \alpha_3! \tag{3.35}$$

$$\partial_{\mathbf{r}'}^{\boldsymbol{\alpha}} f(\mathbf{r}') = \partial_{x'}^{\alpha_1} \partial_{y'}^{\alpha_2} \partial_{z'}^{\alpha_3} f(\mathbf{r}') \tag{3.36}$$

$$(\mathbf{r}' - \mathbf{r})^{\boldsymbol{\alpha}} = (x' - x)^{\alpha_1} (y' - y)^{\alpha_2} (z' - z)^{\alpha_3}. \tag{3.37}$$

We proceed by inserting the expansion of $\rho(\mathbf{r}')$ around $\mathbf{r}$ into (3.27):

$$W(\mathbf{r}, t) = \sum_{k=0}^{\infty} \sum_{\{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=k\}} \frac{\left[\partial_{\mathbf{r}'}^{\boldsymbol{\alpha}} \rho(\mathbf{r}')\right]_{\mathbf{r}'=\mathbf{r}}}{\boldsymbol{\alpha}!} \int_{\mathbb{R}^3} \mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2} (\mathbf{r}' - \mathbf{r})^{\boldsymbol{\alpha}} \mathrm{d}^3 r' \tag{3.38}$$

Using the result [52]

$$\int_{-\infty}^{\infty} \mathrm{e}^{-x^2 t^2} x^n \mathrm{d}x = \begin{cases} \frac{\sqrt{\pi} n!}{2^n (n/2)!} t^{-1-n} & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases} \tag{3.39}$$

and, noting that all terms where any of the components of the multi-index $\boldsymbol{\alpha}$ is odd will integrate to 0, we obtain

$$W(\mathbf{r}, t) = \sum_{k=0}^{\infty} \frac{\pi^{3/2}}{2^{2k}} t^{-(3+2k)} \sum_{\{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=k\}} \frac{\left[\partial_{\mathbf{r}'}^{2\boldsymbol{\alpha}} \rho(\mathbf{r}')\right]_{\mathbf{r}'=\mathbf{r}}}{\boldsymbol{\alpha}!}. \tag{3.40}$$

By identifying that successive applications of the Laplacian operator can be written as

$$\Delta^k \equiv (\nabla^2)^k \equiv (\partial_x^2 + \partial_y^2 + \partial_z^2)^k = \sum_{\{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=k\}} \frac{k!}{\boldsymbol{\alpha}!} \partial_{\mathbf{r}}^{2\boldsymbol{\alpha}}, \tag{3.41}$$

we can further simplify (3.40) into

$$W(\mathbf{r}, t) = \pi^{3/2} \sum_{k=0}^{\infty} \frac{\Delta^k \rho(\mathbf{r})}{2^{2k} k!} t^{-(3+2k)} \approx \pi^{3/2} \left[\frac{\rho(\mathbf{r})}{t^3} + \frac{\Delta \rho(\mathbf{r})}{4t^5} + \mathcal{O}(t^{-7}) \right]. \tag{3.42}$$

For sufficiently large t, the leading term, proportional to t^{-3}, becomes the dominant one.

We have then characterized the function to be integrated in t space in the following manner: for sufficiently small t, the integrand is an even polynomial of t as given by (3.29).

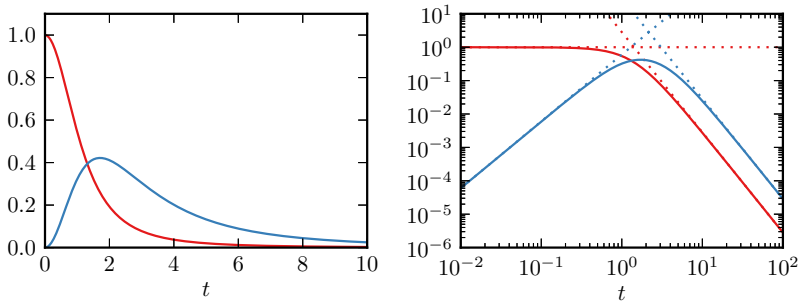


Figure 3.1 $W(0,t)$ for $\rho(\mathbf{r}) = G_2(\mathbf{r})$ ($Q = 1$, blue) and $\rho(\mathbf{r}) = G_{10}(\mathbf{r}) - G_2(\mathbf{r})$ ($Q = 0$, red) in (a) linear and (b) logarithmic coordinates. The leading terms of the small and large t approximations, as given by (3.29) and (3.42), are represented in dotted lines in the right figure.

For large t, it decays proportionally to t^{-3}, as in (3.42). In the region of transition between these two regimes, $W(\mathbf{r},t)$ is a smooth, slowly decaying function. This is illustrated in Fig. 3.1 for a Gaussian charge densities given by

$$G_\alpha(\mathbf{r}) = \left(\frac{\alpha}{\pi}\right)^{3/2} \mathrm{e}^{-\alpha r^2}. \tag{3.43}$$

For these, $W(\mathbf{r},t)$ can be computed analytically as

$$W(\mathbf{r},t) = \left(\frac{\alpha}{\alpha + t^2}\right)^{3/2} \mathrm{e}^{-[\alpha t^2/(\alpha+t^2)]r^2}. \tag{3.44}$$

In general, we cannot assume any closed form for $\rho(\mathbf{r})$, and the integration in t space must therefore be carried out numerically. For that purpose, we resort to one of the most powerful and efficient numerical integration techniques: Gaussian quadrature.

Gaussian quadrature

The word *quadrature* was originally used to denote methods to approximate the areas of geometric figures. In its modern acceptation, a quadrature is a technique where a set of n pairs of points and weights $\{(x_i, \omega_i) | 1 \leq i \leq n\}$, i.e. the *quadrature rule*, is used to approximate a definite integral as

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{p=1}^{n} \omega_p f(x_p) \tag{3.45}$$

Simple quadratures such as the rectangle, trapezoid or Simpson's rule are probably familiar to the reader.

Gaussian quadrature applies to more general integrals of the type

$$\int_a^b w(x) f(x)\mathrm{d}x \approx \sum_{p=1}^{n} \omega_p f(x_p) \tag{3.46}$$

where $w(x) \leq 0 \; \forall \; x \in [a,b]$ is a *weight function*. The quadrature rule depends on $w(x)$ and the interval limits a and b, and is such that (3.46) holds exactly when $f(x)$ is a polynomial of order $\leq 2n-1$.

The points for the n-point rule are the roots of an *orthogonal polynomial* $P_n(x)$ of degree n, which has the following properties:

- It is orthogonal to every polynomial of order $< n$ with respect to $w(x)$ in the interval $[a, b]$, in other words

$$\int_a^b w(x)P_n(x)x^k \mathrm{d}x = 0 \;\forall\; k < n. \tag{3.47}$$

- Its n roots $\{x_i\}$ are distinct and lie in the real interval $[a, b]$, that is, they can be ordered as $a < x_1 < \cdots < x_n < b$.

The orthogonal polynomial $P_n(x)$ depends on $w(x)$, the integration interval $[a, b]$ and the order n. In the most general case, when the integration interval is $[-1, 1]$ and $w(x) = 1$, $P_n(x)$ is the Legendre polynomial of order n, given by the Rodrigues' formula

$$P_n(x) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2 - 1)^n. \tag{3.48}$$

Other classes of polynomials for which tabulated points and weights are commonly found are Chebyshev ($w(x) = (1 - x^2)^{-1/2}$, $[-1, 1]$) and Hermite ($w(x) = \mathrm{e}^{-x^2}$, $(-\infty, \infty)$) polynomials.

For the Gauss-Legendre quadrature, the tabulated points and weights $\{(x_i', \omega_i')\}$ are commonly given in the interval $[-1, 1]$. The integration range can be shifted using a change of variable,

$$\int_a^b f(x)\mathrm{d}x = \frac{2}{b-a}\int_{-1}^{1} f\left(\frac{(b-a)x' + (b+a)}{2}\right)\mathrm{d}x'. \tag{3.49}$$

The tabulated points and weights can be accordingly transformed as

$$x_i = \frac{(b-a)x_i' + (b+a)}{2} \tag{3.50}$$

$$\omega_i = \frac{2\omega_i'}{b-a} \tag{3.51}$$

The formula for the weights can be obtained by constructing an interpolating basis $\{\xi_i(x)\}$ using the roots of $P_n(x)$,

$$\xi_i(x) = \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{(x - x_j)}{(x_i - x_j)} \tag{3.52}$$

which has the following property:

$$\xi_i(x_j) = \delta_{ij}. \tag{3.53}$$

We proceed by approximating the function to be integrated, $f(x)$, by means of an interpolating polynomial:

$$f(x) \approx \sum_i f(x_i)\xi_i(x). \tag{3.54}$$

Integrating the interpolating polynomial over the $[a, b]$ interval yields

$$\int_a^b w(x)f(x)\mathrm{d}x \approx \sum_i f(x_i)\omega_i \tag{3.55}$$

where we define the weights ω_i as

$$\omega_i = \int_a^b w(x)\xi_i(x)\mathrm{d}x. \tag{3.56}$$

In the following, we will prove that using the roots of $P_n(x)$ as points and the values from (3.56) as weights yields a quadrature rule that provides exact results for polynomials of degree degree $\leq 2n-1$.

Let $f(x)$ be a polynomial of degree $\leq 2n-1$, then it can be factorized as

$$f(x) = P_n(x)Q(x) + R(x) \tag{3.57}$$

where both $Q(x)$ and $R(x)$ are of degree $\leq n-1$. Therefore, because of (3.47), we have that $\int_a^b w(x)P_n(x)Q(x)\mathrm{d}x = 0$ and therefore

$$\int_a^b w(x)f(x)\mathrm{d}x = \int_a^b w(x)R(x)\mathrm{d}x \tag{3.58}$$

$R(x)$ is of order $\leq n-1$, and thus it can be exactly represented in terms of the interpolating functions $\xi_i(x)$, so

$$\begin{aligned}\int_a^b w(x)R(x)\mathrm{d}x &= \int_a^b w(x)\left[\sum_i R(x_i)\xi_i(x)\right]\mathrm{d}x \\ &= \sum_i R(x_i)\int_a^b w(x)\xi_i(x)\mathrm{d}x = \sum_i R(x_i)\omega_i.\end{aligned} \tag{3.59}$$

where we have used (3.56).

Because $P_n(x)$ is 0 at its roots, it follows from (3.57) that

$$f(x_i) = R(x_i) \tag{3.60}$$

Combining (3.58), (3.59) and (3.60), we conclude that

$$\int_a^b w(x)f(x)\mathrm{d}x = \sum_i \omega_i f(x_i). \tag{3.61}$$

Quadrature for the Coulomb potential

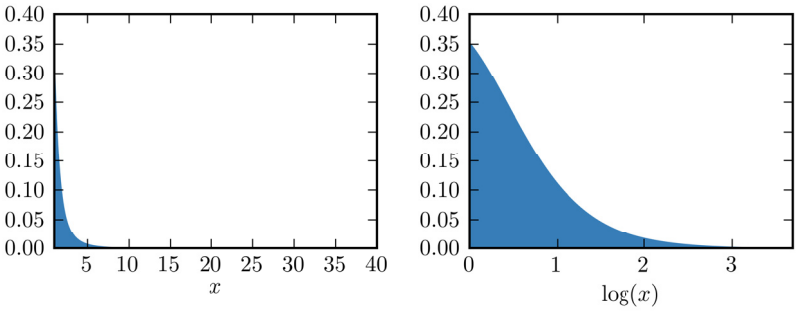
As shown earlier, the integrand $W(\mathbf{r}, t)$ can be subdivided into three distinct regions, which can be accurately integrated using different strategies:

- In the interval $[0, t_{\mathrm{lin}}]$, with $t_{\mathrm{lin}} > 0$, $W(\mathbf{r}, t)$ can be well approximated with a polynomial, as in (3.29). Therefore, the integral over this interval can be accurately approximated using Gauss-Legendre quadrature:

$$\int_0^{t_{\mathrm{lin}}} W(\mathbf{r}, t)\mathrm{d}t \approx \sum_{p=1}^{N_{\mathrm{lin}}} \omega_p W(\mathbf{r}, t_p). \tag{3.62}$$

- The interval $[t_{\mathrm{lin}}, t_{\mathrm{log}}]$, with $t_{\mathrm{log}} > t_{\mathrm{lin}}$, is the transition between the regimes of (3.29) and (3.42). In this interval, $W(\mathbf{r}, t)$ is a slowly decaying function. Hence, the same integration technique as in the interval $[0, t_{\mathrm{lin}}]$ cannot be applied, because the integrand cannot be approximated with a polynomial over a long range. A better approach is integrating the function in *logarithmic* coordinates, using the identity:

$$\int_a^b f(x)\mathrm{d}x = \int_{\log(a)}^{\log(b)} f(\mathrm{e}^s)\mathrm{e}^s\mathrm{d}s. \tag{3.63}$$



(a) $(1+x^2)^{-3/2}$ **(b)** $e^x[1+(e^x)^2]^{-3/2}$

Figure 3.2 Pictorial representation of $\int_1^{40}(1+x^2)^{-3/2}\mathrm{d}x$, in (a) linear and (b) logarithmic coordinates. The areas in blue have identical size, if the different lengths of the x axes are considered.

The adequacy of (3.63) is shown in Fig. 3.2 for the integral $\int_1^{\infty}(1+x^2)^{-3/2}\mathrm{d}x$. Although the areas in both figures are the same, in logarithmic coordinates the curve is much smoother, and therefore easier to fit with a polynomial. In this range, the integral is approximated as

$$\int_{t_{\mathrm{lin}}}^{t_{\mathrm{log}}} W(\mathbf{r},t)\mathrm{d}t \approx \sum_{p=1}^{N_{\mathrm{log}}} \beta_p \mathrm{e}^{\nu_p} W(\mathbf{r},\mathrm{e}^{\nu_p}) \tag{3.64}$$

where $\{(\nu_p,\beta_p)\}$ are the points and weights for a regular Gauss-Legendre quadrature rule in the interval $[\log(t_{\mathrm{lin}}),\log(t_{\mathrm{log}})]$. Notice that integration in logarithmic coordinates yields another quadrature rule, where the points and weights are given by $\{(t_p=\mathrm{e}^{\nu_p},\omega_p=\mathrm{e}^{\nu}_p\beta_p)\}$.

- In the interval $[t_{\mathrm{log}},\infty)$, for a sufficiently large t_{log}, $W(\mathbf{r},t)$ can be accurately represented by the first term in the series in (3.42). Hence we can approximate the integral as

$$\frac{2}{\sqrt{\pi}}\int_{t_{\mathrm{log}}}^{\infty} W(\mathbf{r},t)\mathrm{d}t \approx \frac{2}{\sqrt{\pi}}\int_{t_{\mathrm{log}}}^{\infty} \pi^{3/2}\frac{\rho(\mathbf{r})}{t^3}\mathrm{d}t = \frac{\pi}{t_{\mathrm{log}}^2}\rho(\mathbf{r}). \tag{3.65}$$

The error in this interval is given to first order by

$$\epsilon(\mathbf{r}) \approx \frac{\pi}{8t_{\mathrm{log}}^4}\Delta\rho(\mathbf{r}) + \mathcal{O}(t_{\mathrm{log}}^{-6}) \tag{3.66}$$

In total we have $R = N_{\mathrm{lin}} + N_{\mathrm{log}}$ quadrature points, we can summarize the integration as

$$V(\mathbf{r}) \approx \frac{2}{\sqrt{\pi}}\sum_{p=1}^{R}\omega_p \int_{\mathbb{R}^3} \mathrm{e}^{-t_p^2(\mathbf{r}-\mathbf{r}')^2}\rho(\mathbf{r}')\mathrm{d}^3r' + \frac{\pi}{t_{\mathrm{log}}^2}\rho(\mathbf{r}). \tag{3.67}$$

The expression for the potential in (3.67) can be obtained also by approximating the Coulomb potential as

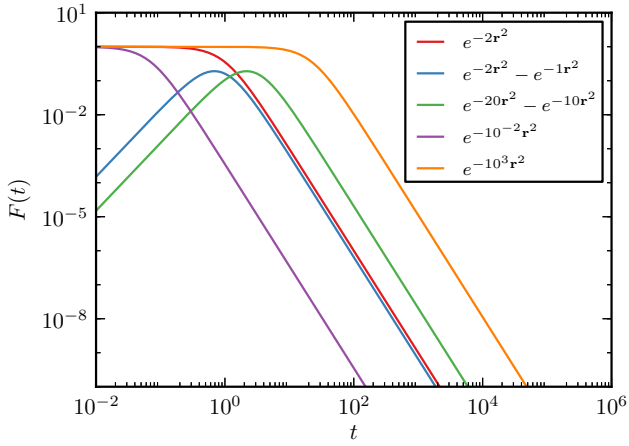


Figure 3.3 $F(t)$ for different Gaussian charge distributions. Normalization constants have been omitted from the legend.

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} \approx \frac{2}{\sqrt{\pi}} \sum_{p=1}^{R} \omega_p \frac{1}{|\mathbf{r}-\mathbf{r}'|} + \frac{\pi}{t_{\text{log}}^2} \delta^3(\mathbf{r}-\mathbf{r}'). \tag{3.68}$$

In other words, by computing the integral up to $t \to \infty$, the singularity of the Coulomb potential has been included explicitly.

The quadrature has four parameters to be chosen: N_{lin}, N_{log}, t_{lin} and t_{log}. In general lines, the accuracy of the quadrature can be improved by increasing t_{log}, N_{lin} and N_{log}. t_{log} is the ultimately limiting factor, and should be chosen so that the error caused by truncating the tail given by (3.66) is smaller than the desired maximum error. If the charge density presents very steep regions where $\nabla^2\rho(\mathbf{r})$ is large, t_{log} will likewise need to be large. However, t_{log} should be kept as small as possible, as a larger integration range will require a larger N_{log} to yield the same accuracy. An adequate choice of t_{lin} minimizes the total number of quadrature points needed. (3.42) suggests that if long range interactions are important, due to large accumulations of charge far apart from each other, $W(\mathbf{r},t)$ can have a very complicated structure close to $t = 0$.

The choice of t_{lin} and t_{log} must be adequate for every point in space. One tool to inspect the "average" form of the integrand $W(\mathbf{r},t)$, is the function

$$F(t) = \int_{\mathbb{R}^3} \rho(\mathbf{r}) W(\mathbf{r},t) \mathrm{d}^3 r \tag{3.69}$$

which is the expansion of the self-interaction energy in t-space, i.e.

$$E = \int_{\mathbb{R}^3} \rho(\mathbf{r}) V(\mathbf{r}) \mathrm{d}^3 r = \frac{2}{\sqrt{\pi}} \int_0^\infty F(t) \mathrm{d}t. \tag{3.70}$$

For a Gaussian charge distribution with exponent α given in (3.43), $F(t)$ is given by

$$F(t) = \left(\frac{\alpha}{\alpha + 2t^2}\right)^{3/2}. \tag{3.71}$$

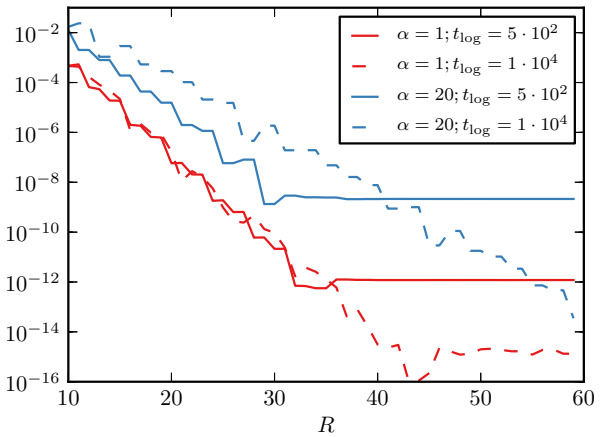


Figure 3.4 Error in the evaluation of $\frac{2}{\sqrt{\pi}} \int_0^\infty \left(\frac{\alpha}{\alpha + t^2}\right)^{3/2} \mathrm{d}t$ as a function of the number of quadrature points, for different values of $t_{\log}$. t_{lin} is fixed in all cases to 2. The amount of quadrature points in each interval is set such that $N_{\text{lin}} = \lfloor R/2 \rfloor$ and $N_{\log} = \lceil R/2 \rceil$.

In Fig. 3.3, $F(t)$ is illustrated for different charge distributions. As discussed earlier, for small t, $F(t)$ is either constant of proportional to t^2, and $F(t=0)$ is the total charge (0 or 1 in this case). For large t, $F(t)$ is proportional to t^{-3} for all systems, although the transition from the polynomial to the decaying regime occurs at different values of t. The quadrature parameters t_{lin} and $t_{\log}$ should be chosen accordingly.

The accuracy of the quadrature is demonstrated in Fig. 3.4, where the function $F(t)$ given in (3.71) is integrated for two different exponents of 1 and 20. The value of t_{lin} is fixed to 2, and two values of $t_{\log}$, 500 and 10^4, are tested. The total amount of quadrature points is evenly split between the linear and logarithmic intervals, such that $N_{\text{lin}} = \lfloor R/2 \rfloor$ and $N_{\log} = \lceil R/2 \rceil$. As it can be seen, the maximum accuracy is limited by $t_{\log}$, but ultimately, sufficiently large values of $t_{\log}$ and R are able to provide any desired accuracy. A larger $t_{\log}$, however, requires a larger amount of quadrature points, as is clearly visible for $\alpha = 20$.

3.4 The Helmholtz kernel

Another partial differential equation of importance in physics is the inhomogeneous Helmholtz equation:

$$(\nabla^2 + \kappa^2) f(\mathbf{r}) = -4\pi g(\mathbf{r}) \tag{3.72}$$

where κ is a constant. Notice that Poisson's equation (3.3) is a particular case of (3.72) for $\kappa = 0$. The Green's function for (3.72) is

$$G(\mathbf{r}; \mathbf{r}') = \frac{\mathrm{e}^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}. \tag{3.73}$$

This function is referred to as the screened Coulomb potential, the Debye-Hückel potential [53] or the Yukawa potential [54] in different contexts.

Let us compare (3.72) with the time-independent Schrödinger equation for a single particle moving in an external potential $V(\mathbf{r})$

$$-\frac{1}{2}\nabla^2\psi(\mathbf{r})+V(\mathbf{r})\psi(\mathbf{r})=\epsilon\psi(\mathbf{r}). \tag{3.74}$$

Equation (3.74) can be rewritten in an integral form [55, 56]:

$$\psi(\mathbf{r})=\frac{-1}{2\pi}\int_{\mathbb{R}^3}\frac{\mathrm{e}^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}V(\mathbf{r}')\psi(\mathbf{r}')\mathrm{d}^3r'. \tag{3.75}$$

with

$$\kappa=\sqrt{-2\epsilon} \tag{3.76}$$

In the rest of this discussion, we assume bound states for which $\epsilon<0$ and correspondingly $\kappa>0$. SCF algorithms based on this approach have been successfully implemented [57–60].

Similarly to the Laplace expansion of the Coulomb potential in (3.10), the Helmholtz kernel can be expressed in terms of the spherical harmonics as [53]

$$\frac{\mathrm{e}^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}=8\kappa\sum_{l=0}^{\infty}\frac{4\pi}{2l+1}\sum_{m=-l}^{l}Y^*_{lm}(\theta,\phi)Y_{lm}(\theta',\phi')\hat{I}_{l+\frac{1}{2}}(\kappa r_<)\hat{K}_{l+\frac{1}{2}}(\kappa r_>) \tag{3.77}$$

$\hat{I}_i(x)$ and $\hat{K}_i(x)$ are respectively the modified spherical Bessel functions of the first and second kind.*

For a function $g(\mathbf{r})$ in the form $\sum_{lm}g^{lm}(r)Y_{lm}(\theta,\phi)$, its convolution with the Helmholtz kernel $f(\mathbf{r})$ can be expanded in the same manner, and the corresponding radial functions are given by

$$f^{lm}(r)=\frac{32\pi\kappa}{2l+1}\left[\hat{K}_{l+\frac{1}{2}}(\kappa r)\int_0^r g^{lm}(s)\hat{I}_{l+\frac{1}{2}}(\kappa s)s^2\mathrm{d}s+\hat{I}_{l+\frac{1}{2}}(\kappa r)\int_r^{\infty}g^{lm}(s)\hat{K}_{l+\frac{1}{2}}(\kappa s)s^2\mathrm{d}s\right] \tag{3.78}$$

It is also possible to rewrite the Helmholtz kernel using a Cartesian-separable integral identity akin to (3.24)

$$\frac{\mathrm{e}^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}=\frac{2}{\sqrt{\pi}}\int_0^{\infty}\mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2-\kappa^2/4t^2}\mathrm{d}t. \tag{3.79}$$

In this form, it is also possible to compute the integral in (3.75) using a quadrature scheme similar to one presented in Section 3.3. Let us write

$$\psi(\mathbf{r})=\frac{2}{\sqrt{\pi}}\int_0^{\infty}\mathrm{e}^{-\kappa^2/4t^2}Q(\mathbf{r},t)\mathrm{d}t \tag{3.80}$$

where $Q(\mathbf{r},t)$ is

$$Q(\mathbf{r},t)=\int_{\mathbb{R}^3}\mathrm{e}^{-t^2(\mathbf{r}-\mathbf{r}')^2}V(\mathbf{r}')\psi(\mathbf{r}')\mathrm{d}^3r'\mathrm{d}t \tag{3.81}$$

similarly to the $W(\mathbf{r},t)$ defined in (3.26),

For sufficiently large t, the factor $\mathrm{e}^{-\kappa^2/4t^2}$ is practically 1, and $Q(\mathbf{r},t)$ becomes identical to $W(\mathbf{r},t)$. Hence, the large-t behaviour will be as given by (3.42), changing $\rho(\mathbf{r})$ by $V(\mathbf{r})\psi(\mathbf{r})$. For small values of t, the picture changes though, because at $t=0$ the factor $\mathrm{e}^{-\kappa^2/4t^2}$ is flat, i.e., all its derivatives are 0. Therefore, the function is not analytic and cannot be represented by a Taylor series. On the other hand, precisely because very close to

*In [52] $\hat{K}_i(x)$ is referred to as modified Bessel function of the third kind.

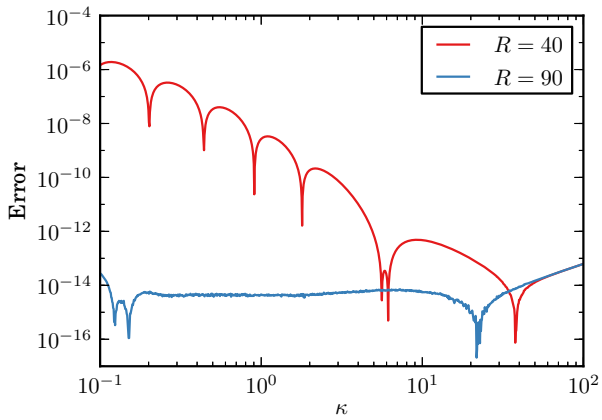


Figure 3.5 Absolute error in the numerical integration of (3.82).

the origin $t = 0$ $e^{-\kappa^2/4t^2}$ is almost 0, there exists some $t_{\min} > 0$ such that the integral in the interval $[0, t_{\min}]$ can be neglected.

In Fig. 3.5, the error in the numerical integration of

$$\int_0^\infty e^{-\kappa^2/4x^2} \left(1+x^2\right)^{-3/2} dx = 1 - \frac{k\sqrt{\pi}}{2} e^{\kappa^2/4} \mathrm{erfc}(k/2) \tag{3.82}$$

is shown. The quadrature parameters are

$$t_{\min} = \frac{k\sqrt{-\log(10^{-12})}}{2} \tag{3.83}$$

which is chosen so that $e^{-\kappa^2/4t^2} \leq 10^{-12}$ for $t \in [0, t_{\min}]$, and

$$t_{\mathrm{lin}} = \sqrt{\kappa \left(\kappa + \sqrt{\frac{\kappa^2 + 24}{12}} \right)} \tag{3.84}$$

which is the point where the integrand in (3.82) has its maximum. The end point $t_{\log}$ is fixed at 10^4. The scheme shows that it is possible to obtain an accurate result for all values of κ in the range $[10^{-1}, 10^2]$. However, smaller values of κ, corresponding to smaller energies in absolute value, require a larger amount of quadrature points to be treated accurately. More consistent and affordable integration schemes, obtained by e.g. some change of variable, would be desired for actual calculations.

4 Representation of three-dimensional scalar functions

In the previous sections, we reviewed the equations needed to solve, in an approximate fashion, the electronic time-independent Schrödinger equation for molecular systems. Let us suppose that we have settled for one of the methods that should be adequate to model a certain chemical problem. As the equations are in general very complicated, we would want to set up a computer program to solve them. Modern digital computers work in terms of finite amounts of digits and arithmetic operations. Hence, the first step is the *discretisation* of our problem. In other words, we need to translate our equations into a collection of floating-point numbers.

The first point to be addressed is how to represent the wave function. In most of the methods presented in the previous section, wave functions were expressed in terms of molecular orbitals. Hence, in general terms, the question to be answered is *"how do we represent three-dimensional scalar functions using just a list of numbers?"*. Whatever our choice, it must be done judiciously, as it will ultimately determine the efficiency and accuracy of the solutions.

In this chapter, we will focus on two different discretisation strategies, and the fundamental operations required to implement algorithms for electronic structure calculations, as presented in Chapter 2. First, we will discuss atom-centred basis sets, with a particular focus on Gaussian-type orbitals, which are, by far, the most common type of basis set found in electronic structure calculations. Second, we will present the *bubbles* representation, which was introduced in Publication II and is one of the main points addressed in this thesis.

4.1 The basis set approximation

In general, a function can be discretised as a linear combination of certain known *basis functions*:

$$f(\mathbf{r}) = \sum_{p=1}^{N} c_p \chi_p(\mathbf{r}). \tag{4.1}$$

The set of functions $\{\chi_i(\mathbf{r})\}$ is the *basis set*. The basis set is said to be complete in a given set (such as e.g. L^2, the set of all square-integrable functions) if (4.1) holds exactly for all functions belonging to that set. Although many such basis sets exist for L^2, they are infinite. In practical calculations, finite, incomplete basis sets must be used and hence (4.1) is approximate.

Once the basis set is fixed, the representation of the function is fully determined by the expansion coefficients $\{c_p\}$. The solutions of the time-independent Schrödinger equation

can be chosen to be real for real scalar potentials [61], and thus we will concern ourselves exclusively with real basis sets and expansion coefficients.

Ideally, we wish the following properties from a basis set:

- **Accuracy**

  The basis must be able to represent the target functions faithfully, and provide results that are sufficiently accurate for a given purpose.

- **Compactness**

  For a given accuracy, the size of the basis set should be as small as possible.

- **Efficiency**

  The operations involving the basis functions should be performed as fast as possible.

- **Systematicity**

  The basis set should depend on a set of parameters that can be modified such that the accuracy of a given calculation will improve.

- **Universality**

  The performance, in terms of accuracy and efficiency, should be adequate to model a large variety of properties and systems.

To devise a basis set that will satisfy as many of the requested features at possible, we shall first examine the analytical properties of the exact wave function.

Properties of the exact ground state wave function

The exact ground state wave function is continuous, and its first derivatives are continuous except at the points where the Coulomb potential is singular. At the nuclear positions, the wave function fulfils the following condition [62]:

$$\left.\frac{\partial \psi}{\partial |\mathbf{r}_i - \mathbf{R}_A|}\right|_{\mathbf{r}_i=\mathbf{R}_A} = -Z_A \psi|_{\mathbf{r}_i=\mathbf{R}_A}. \tag{4.2}$$

This means that, when electron 1 is in the vicinity of nucleus A ($\mathbf{r}_1 \approx \mathbf{R}_A$), the wave function presents a cusp of the form

$$\psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) \sim \mathrm{e}^{-\zeta|\mathbf{r}_1 - \mathbf{R}_A|} \psi(\mathbf{R}_A, \mathbf{r}_2, \ldots, \mathbf{r}_N) \tag{4.3}$$

for some constant ζ. The first derivatives of the wave function are discontinuous at the nuclear positions. The electron density presents similar nuclear cusps,

$$\rho(\mathbf{r}) \sim \mathrm{e}^{-2Z_k|\mathbf{r}-\mathbf{R}_k|}. \tag{4.4}$$

At long distances, i.e., $\mathbf{r} \gg \mathbf{R}_k \forall k$, the density decays exponentially as [63, 64]

$$\rho(\mathbf{r}) \sim \mathrm{e}^{-2\sqrt{2I}|\mathbf{r}-\mathbf{R}_k|}, \tag{4.5}$$

where I is the ionisation potential.

Let us consider now a wave function of the form of a Slater determinant. From the above properties, the molecular orbitals $\phi_i(\mathbf{r})$ can be characterized as three-dimensional functions, smooth at every point in space except at the nuclear positions, where their first derivatives are discontinuous. This leads to very sharp cusps at some of the nuclear positions. At long distances, the molecular orbitals decay exponentially, with an exponent of $\sqrt{2I}$ or faster. Similar properties can be ascribed to products of two molecular orbitals.

Other three-dimensional scalar functions which will be required in some cases are electrostatic potentials. From Poisson's equation (3.3), the charge density is the Laplacian of

	l		
m	0	1	2
-2			$\sqrt{3}\hat{x}\hat{y}$
-1		$\hat{y}$	$\sqrt{3}\hat{y}\hat{z}$
0	1	$\hat{z}$	$\frac{1}{2}(2\hat{z}^2-\hat{x}^2-\hat{y}^2)$
1		$\hat{x}$	$\sqrt{3}\hat{x}\hat{z}$
2			$\frac{\sqrt{3}}{2}(\hat{x}^2-\hat{y}^2)$

Table 4.1 Real-valued spherical harmonics $Y_{lm}(\theta,\phi)$ up to order $l=2$.

the potential. Hence, for a charge distribution continuously differentiable to order k, the potential is continuously differentiable up to order $k+2$. For charge distributions given by products of molecular orbitals, the potential is continuously differentiable to at least second order. At long distances, the potential decays as r^{-k} for some $k>0$, as in the multipole expansion in (3.15).

An analytic test case: the hydrogen-like atom

The hydrogen-like atom consists of one electron and a nucleus of charge Z. It is the simplest system consisting of two particles which interact electrostatically, for which the Schrödinger equation can be solved analytically. The hydrogen-like atom can be used as a starting point to construct a basis set as, quite obviously, the hydrogen-like functions must be accurately represented in any basis set in which we intend to expand the more complicated molecular orbitals. The bounded eigenfunctions of the hydrogen-like atom in spherical coordinates are given by [28]

$$\psi_{nlm}(\mathbf{r})=R_{nl}(r)Y_{lm}(\theta,\phi). \tag{4.6}$$

The radial functions $R_{nl}(r)$ are

$$R_{nl}(r)=N_{nl}\left(2a_n r\right)^l L_{n-l-1}^{2l+1}\left(2a_n r\right)\mathrm{e}^{-a_n r}, \tag{4.7}$$

where $a_n=Z/n$, N_{nl} is a normalization constant, and the $L_j^k(x)$ functions are the associated Laguerre polynomials. The angular functions $Y_{lm}(\theta,\phi)$ are the aforementioned spherical harmonics*, which can be written in terms of the components of the unit vector $\hat{\mathbf{r}}=\mathbf{r}/r=(x/r,y/r,z/r)=(\hat{x},\hat{y},\hat{z})$:

$$Y_{lm}(\theta,\phi)=\sum_{uvw}C_{uvw}^{lm}\hat{x}^u\hat{y}^v\hat{z}^w \tag{4.8}$$

The values of the coefficients C_{uvw}^{lm} can be obtained from recursive formulas [28]. The explicit expressions for the real-valued spherical harmonics of order up to $l=2$ are given in Table 4.1.

The 1*s* orbital is

$$\psi(\mathbf{r})=\sqrt{\frac{Z^3}{\pi}}\mathrm{e}^{-Zr}, \tag{4.9}$$

which fulfils (4.2). Likewise, the electron density,

*In this work, we will refer exclusively to the real-valued spherical harmonics, which are linear combinations of the spherical harmonics proper.

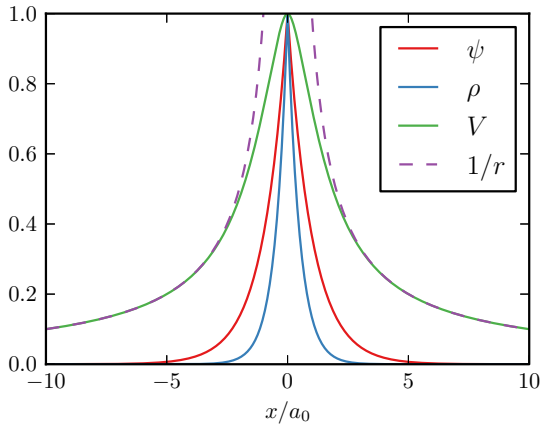


Figure 4.1 Wave function, electron density and electrostatic potential for the ground state of the hydrogen atom along the x axis. The functions have been normalized to 1 at x=0. The nuclear potential (with the sign changed) has been superimposed in discontinuous lines.

$$\rho(\mathbf{r}) = |\psi(\mathbf{r})|^2 = \frac{Z^3}{\pi} e^{-2Zr}, \tag{4.10}$$

decays as given by (4.5), as the ionization potential is $I = Z^2/2$. The electrostatic potential caused by the electron distribution of a hydrogen atom can be solved analytically by means of (3.14):

$$V(\mathbf{r}) = \frac{1-(1+Zr)\,e^{-2Zr}}{r}, \tag{4.11}$$

which, for sufficiently large distances, becomes the potential of a unit point charge $1/r$.

The three functions ψ, ρ and V are depicted in Fig. 4.1 along the x axis. As it can be seen, the singularity of nuclear potential causes the electronic wave function to become steep. This is reflected as well in the electron density and the electrostatic potential.

4.2 Atom-centred basis functions

In a molecule, the presence of various nuclei breaks the angular symmetry of the hydrogen-like atomic eigenfunctions. However, the nuclear potential is so steep that, in the vicinity of each nucleus, the spherical symmetry is retained to a large extent. In other words, molecules can be regarded to a good approximation as a combination of atoms. This idea agrees well with the traditional chemical doctrine, where molecules are seen as collections of atoms linked by bonds. Serious attempts to reconcile this chemical notion with quantum mechanics have been attempted [65].

A reasonable approach is then to construct the molecular orbitals as linear combinations of atomic orbitals, that is, functions which have the same symmetry as the hydrogen-like eigenfunctions in (4.6)

$$\chi_p(\mathbf{r}) = R_p^{\mathrm{AO}}(r) Y_{l_p m_p}(\theta, \phi). \tag{4.12}$$

For a molecule with K atoms, each atom A, located at $\mathbf{R}_A = (X_A, Y_A, Z_A)$, is assigned a set of atomic orbitals, $B_A = \{\chi_p\}$. Molecular orbitals are then expanded as

$$\phi_i(\mathbf{r}) = \sum_{A=1}^{K} \sum_{p|\ \chi_p \in B_A} c_{ip} \chi_p(\mathbf{r}_A) \tag{4.13}$$

The relative coordinates with respect to nucleus A are denoted $\mathbf{r}_A = \mathbf{r} - \mathbf{R}_A$.

A *minimal* basis set contains only the basis functions corresponding to the shells which are occupied in the isolated gas-phase atom. The basis set can be made more complete in two ways: by increasing the maximum allowed angular momentum number of the basis set, L, and by adding more functions with different radial parts. The new radial functions are often obtained by some optimization procedure. For SCF calculations, the energy converges fast with L, as e^{-L}, so a very large L is typically not needed. On the contrary, for correlated calculations, it is critical to use a basis set with sufficiently large L, as the correlation energy converges much more slowly, as $(L+1)^{-3}$ [66–68].

There are several possible choices for the actual form of the radial functions, each spanning a different type of basis sets. A natural choice for the atomic orbitals are the eigenstates of the hydrogen-like atom (composed of one electron and one nucleus of charge Z), as given in (4.7). Although the hydrogen-like orbitals provide a qualitatively reasonable description for small systems, in practice they are a poor choice for many-electron calculations. The reason is the limited choice of functions with the same l for a given element. Moreover, the unbounded continuum states must be also added to have a complete basis set.

Slater-type orbitals (STOs) [69] constitute a more efficient approach. Instead of being few functions with complicated nodal structures, as is the case of the hydrogen-like atom eigenfunctions, Slater-type orbital basis sets are composed of simpler, nodeless radial functions with a larger variety of exponents:

$$R_p^{\mathrm{STO}}(r) = N_p r^{n_p} \mathrm{e}^{-\zeta_p r} \tag{4.14}$$

The set of exponents $\{\zeta_p\}$ is obtained by some optimization procedure, with respect to e.g. the atomic ground state energy. Slater-type orbitals often provide accurate approximations to the wave function. Nevertheless, multi-centre integrals of the type of (2.15) and (2.16) are difficult to compute. An efficient alternative are Gaussian-type orbitals (GTOs) [70]:

$$R_p^{\mathrm{GTO}}(r) = N_p r^{n_p} \mathrm{e}^{-\zeta_p r^2} \tag{4.15}$$

Typically, GTOs are given in a contracted form

$$R_p^{\mathrm{cGTO}}(r) = r^{n_p} \sum_j a_{pj} \mathrm{e}^{-\zeta_{pj} r^2} \tag{4.16}$$

where the coefficients a_{pj} are fixed. A molecular orbital represented in terms of GTOs cannot possibly fulfil the properties of the exact wave function described earlier. However, the tremendous ease to compute integrals over GTOs permits using large basis sets. Because of this, GTOs often yield more accurate results than a calculation with a similar cost using STOs. The efficiency of GTOs in the calculation of two-electron Coulomb integrals will be discussed later.

It is also possible to use more general *numerical* radial functions, where $R_p^{\mathrm{AO}}(r)$ is tabulated at different values of r [71–73]. Typically, these basis sets provide even superior accuracy, although integrals are likewise expensive.

The Roothaan–Hall equations

The Hartree–Fock equations (2.21) can be rewritten in terms of the AO basis, by expanding the molecular orbitals in (4.13). For a closed-shell system, this yields the Roothaan–Hall equations [74,75], which in matrix form can be written as:

$$\mathbf{Fc} = \mathbf{Sc}\boldsymbol{\epsilon}, \tag{4.17}$$

where $\mathbf{c}$ is the matrix containing the molecular orbital coefficients, the diagonal matrix $\boldsymbol{\epsilon}$ contains the real eigenvalues

$$\epsilon_{pq} = \delta_{pq}\epsilon_p. \tag{4.18}$$

$\mathbf{S}$ is the overlap matrix

$$S_{pq} = \int \chi_p^*(\mathbf{r}_1)\chi_q(\mathbf{r}_1)\,\mathrm{d}^3 r_1. \tag{4.19}$$

The Fock matrix $\mathbf{F}$,

$$F_{pq} = h_{pq} + \sum_r^N \sum_s^N D_{rs}\left[g_{pqrs} - \frac{1}{2}g_{prqs}\right] \tag{4.20}$$

is given in terms of the one-electron integrals

$$h_{pq} = \int \chi_p^*(\mathbf{r}_1)\left[-\frac{1}{2}\nabla_1^2 - \sum_A \frac{Z_A}{|\mathbf{r}_1 - \mathbf{R}_A|}\right]\chi_q(\mathbf{r}_1)\,\mathrm{d}^3 r_1 \tag{4.21}$$

and the two-electron integrals

$$g_{pqrs} = \int\int \chi_p(\mathbf{r}_1)\chi_q(\mathbf{r}_1)\frac{1}{r_{12}}\chi_r(\mathbf{r}_2)\chi_s(\mathbf{r}_2)\,\mathrm{d}^3 r_1 \mathrm{d}^3 r_2. \tag{4.22}$$

Note that the Kohn–Sham equations can be also rewritten as (4.17), by using the form of the Fock operator given in (2.25). The density matrix is given in terms of the molecular orbital coefficients:

$$D_{pq} = \sum_{i=1}^N \eta_i c_{pi} c_{qi}. \tag{4.23}$$

In (4.23), η_i is the occupation number of the i-th orbital, e.g. $\eta_i = 2$ for every occupied orbital in a closed-shell system. The electron density depends on the density matrix as

$$\rho(\mathbf{r}) = \sum_{i=1}^N |\phi_i(\mathbf{r})|^2 = \sum_{pq} D_{pq}\chi_p(\mathbf{r}_p)\chi_q(\mathbf{r}_q). \tag{4.24}$$

Despite having the appearance of a generalized eigenvalue problem, the Roothaan–Hall equations are non-linear, as the Fock matrix depends on the molecular orbital coefficients. Hence, it must be solved iteratively. In general, converge cannot be expected from the straightforward iteration of (4.17). More sophisticated convergence acceleration techniques, such as the *direct inversion of the iterative subspace* (DIIS) [76], are commonplace in nowadays standard SCF implementations. Some complicated cases might require more robust optimization techniques [77,78].

The first bottleneck in terms of computing effort is the calculation of an enormous amount of two-electron integrals. For a basis set consisting of N basis functions, there are approximately N^4 integrals. The number can be reduced by about a factor of 8 by realizing that

$$g_{pqrs} = g_{qprs} = g_{pqsr} = g_{qpsr} = g_{rspq} = g_{srpq} = g_{rsqp} = g_{srqp}. \tag{4.25}$$

However, this does not reduce the scaling of the number of integrals with the basis set size. Similar considerations can be applied if the symmetry of the system is exploited. Although the integrals can be stored on disk and reused at every iteration, for a few hundreds

of basis functions the amount of storage required makes this approach impractical. In the alternative on-the-fly approach, the integrals are computed anew every iteration, and then discarded after they have been contracted with the density matrix. This is known as the *direct SCF* method [79]. Furthermore, the use of pre-screenings to avoid computing negligible integrals can drastically reduce the amount of integrals needed [80, 81].

The second major bottleneck is the diagonalization of the Fock matrix. The computational cost of this operation grows as $\mathcal{O}(N^3)$, which becomes unfeasible for large systems. A number of methods exist that directly optimize the density matrix, completely avoiding the diagonalization step [82, 83].

Coulomb integrals over Gaussian-type orbitals

Being one of the most pressing bottlenecks in electronic structure calculations, more so since the dawn of direct SCF methods, the development of efficient two-electron integral codes has been an active area of research in quantum chemistry. As mentioned above, the main reason to use Gaussian-type orbitals is the ease of computation of multi-centre integrals. The first critical advantage of Gaussian-type orbitals is that using (4.8) the spherical forms given by (4.12) and (4.15) can be written as linear combinations of Cartesian-separable functions, usually referred to as *Cartesian* Gaussian-type orbitals:

$$\chi_p(\mathbf{r}) = N_p\, r^{l_p} \mathrm{e}^{-\zeta_p r^2} \hat{x}^{l^x_p} \hat{y}^{l^y_p} \hat{z}^{l^z_p} = N_p (x^{l^x_p} \mathrm{e}^{-\zeta_p x^2}) (y^{l^y_p} \mathrm{e}^{-\zeta_p y^2}) (z^{l^z_p} \mathrm{e}^{-\zeta_p z^2}) \tag{4.26}$$

with $l_p = l^x_p + l^y_p + l^z_p$. The normalization constant is given by $N_p = N^0_p \gamma(l^x_p)\gamma(l^x_p)\gamma(l^x_p)$

$$N^0_p = \pi^{-3/4} 2^{l_p + 3/4} \zeta_p^{(2l_p+3)/4} \tag{4.27}$$

and

$$\gamma(i) = \frac{1}{\sqrt{(2i-1)!!}}. \tag{4.28}$$

The second property is that the product of two Gaussian functions is another Gaussian function. This is known as the Gaussian product rule. For a pair of Cartesian Gaussian-type orbitals, we can write

$$\begin{aligned}
&\left(x^{l^x_p}_p y^{l^y_p}_p z^{l^z_p}_p e^{-\zeta_p r^2_p}\right) \cdot \left(x^{l^x_q}_q y^{l^y_q}_q z^{l^z_q}_q e^{-\zeta_q r^2_q}\right) \\
&\qquad = K_{pq} \left[\sum_{i=0}^{l^x_p + l^x_q} T^{P,x}_{l^x_p l^x_q i} x^i_P\right] \left[\sum_{i=0}^{l^y_p + l^y_q} T^{P,y}_{l^y_p l^y_q i} y^i_P\right] \left[\sum_{i=0}^{l^z_p + l^z_q} T^{P,z}_{l^z_p l^z_q i} z^i_P\right] e^{-\alpha_P r^2_P}.
\end{aligned} \tag{4.29}$$

The coordinates of the new expansion centre are $\mathbf{R}_P = (\zeta_p \mathbf{R}_p + \zeta_q \mathbf{R}_q)/(\zeta_p + \zeta_q)$, and the new exponent is $\alpha_P = \zeta_p + \zeta_q$. The coefficients $T^{P,\xi}_{l^\xi_p l^\xi_q i}$ can be computed from (4.29) using the binomial theorem. The pre-exponential factor K_{pq} is

$$K_{pq} = \mathrm{e}^{-\zeta_p \zeta_q / (\zeta_p + \zeta_q) |\mathbf{R}_q - \mathbf{R}_p|^2}. \tag{4.30}$$

For example, a four-centre two-electron integral over s ($l = 0$) GTOs is given by

$$g_{pqrs} = N_p N_q N_r N_s \frac{\pi^3 K_{pq} K_{rs}}{\left(\alpha_P \alpha_Q\right)^{3/2}} \frac{\mathrm{erf}\left(\sqrt{U_{PQ}} R_{PQ}\right)}{R_{PQ}} \tag{4.31}$$

where

$$U_{PQ} = \alpha_P \alpha_Q / (\alpha_P + \alpha_Q) \tag{4.32}$$

and $R_{PQ} = |\mathbf{R}_Q - \mathbf{R}_P|$, α_Q and $\mathbf{R}_Q$ being defined for the pair rs as α_P and $\mathbf{R}_P$ for the pair pq.

For integrals over function of larger l, there are a number of different schemes available. The integrals over high angular momentum function are obtained by means of recursive relations, such as in the McMurchie–Davidson [48] and the Obara–Saika [49, 50] schemes. Both of these are based on evaluation of the Boys function [70]:

$$F_n(x) = \int_0^1 \mathrm{e}^{-xt^2} t^{2n} \mathrm{d}t. \tag{4.33}$$

The evaluation of the Boys function can be bypassed by means of the Rys-Gauss quadrature [51]. For heavily contracted basis functions, more efficient schemes are available [84, 85]. Variations of the methods above [86, 87] are the basis for some of the most efficient implementations available.

Numerical calculation of two-electron integrals: the *sivari* method

Inserting the integral identity for the Coulomb operator (3.24) in the expression for the two electron integrals (4.22), the following expression is obtained:

$$g_{pqrs} = \frac{2}{\sqrt{\pi}} K_{pq} K_{rs} N^0_{pqrs} \int_0^\infty M_{PQ}(t) \Theta^x_{l^p_x l^q_x l^r_x l^s_x}(t) \Theta^y_{l^p_y l^q_y l^r_y l^s_y}(t) \Theta^z_{l^p_z l^q_z l^r_z l^s_z}(t) \mathrm{d}t. \tag{4.34}$$

The function $M_{PQ}(t)$ is

$$M_{PQ}(t) = \frac{\pi^3 e^{-\Lambda(U_{PQ},t)R^2_{PQ}}}{[(U_{PQ}+t^2)(\alpha_P+\alpha_Q)]^{3/2}} \tag{4.35}$$

where $\Lambda(\alpha, t)$ is given by

$$\Lambda(\alpha, t) = \frac{\alpha t^2}{\alpha + t^2}. \tag{4.36}$$

The four-index tensors $\mathbf{\Theta}^\xi$ in (4.34) are computed using the contraction

$$\Theta^\xi_{l^p_\xi l^q_\xi l^r_\xi l^s_\xi}(t) = \sum_{i=0}^{l^\xi_p + l^\xi_q} \sum_{j=0}^{l^\xi_r + l^\xi_s} \tilde{T}^{P,\xi}_{l^\xi_p l^\xi_q i} \Phi_{ij}(t; \Xi_{PQ}) \tilde{T}^{Q,\xi}_{l^\xi_r l^\xi_s j}, \tag{4.37}$$

for ξ one of x, y or z and correspondingly Ξ being X, Y or Z. The coefficients $\tilde{T}_{l^p_\xi l^q_\xi i}$ are the coefficients $T_{l^p_\xi l^q_\xi i}$ from (4.29) multiplied with the γ factors (4.28) of the normalization constants:

$$\tilde{T}^{P,\xi}_{l^p_\xi l^q_\xi i} = \gamma(l^p_\xi)\gamma(l^q_\xi) T^{P,\xi}_{l^p_\xi l^q_\xi i}. \tag{4.38}$$

The functions $\Phi_{ij}(t; \Xi_{PQ})$ are polynomials in Ξ_{PQ}, containing either even or odd powers of Ξ_{PQ}. The first polynomial is given by $\Phi_{00} = 1$ and the higher order Φ_{ij} can be obtained with the following recursive expressions:

$$\Phi_{i+1,j} = \frac{1}{2\alpha_P} \left[\frac{\partial}{\partial \Xi_{PQ}} \Phi_{ij} - 2\Lambda(U_{PQ}, t) \Xi_{PQ} \Phi_{ij} + i \Phi_{i-1,j} \right] \tag{4.39}$$

$$\Phi_{i,j+1} = \frac{1}{2\alpha_Q} \left[-\frac{\partial}{\partial \Xi_{PQ}} \Phi_{ij} + 2\Lambda(U_{PQ}, t) \Xi_{PQ} \Phi_{ij} + j \Phi_{i,j-1} \right]. \tag{4.40}$$

The integration in t in (4.34) can be carried out using the quadrature described in Section 3.3, such that

$$g_{pqrs} \approx \sum_k \omega_k \tilde{g}_{pqrs}(t_k), \tag{4.41}$$

$$\tilde{g}_{pqrs}(t) = \frac{2}{\sqrt{\pi}} \int \int \chi_p(\mathbf{r}_1)\chi_q(\mathbf{r}_1) \mathrm{e}^{-t^2 r_{12}^2} \chi_r(\mathbf{r}_2)\chi_s(\mathbf{r}_2)\, \mathrm{d}^3 r_1 \mathrm{d}^3 r_2. \tag{4.42}$$

The integrals are actually performed along the modified dimension t'

$$t' = \left(U_{PQ}^{-1/2} + R_{PQ} \right) t. \tag{4.43}$$

This enables computing the integrals over all basis functions with practically the same accuracy, regardless of the exponents and the centres of the basis functions. The advantage of this is that the two-electron contribution to the Fock matrix can be written as

$$F_{pq} = h_{pq} + \sum_k \omega_k G_{pq}(t_k), \tag{4.44}$$

where $G_{pq}(t)$ is

$$G_{pq}(t) = \sum_r^N \sum_s^N D_{rs} \left[\tilde{g}_{pqrs} - \frac{1}{2} \tilde{g}_{prqs} \right]. \tag{4.45}$$

This approach is attractive from a computational point of view because it breaks the two electron integrals into simpler parts. The only information that needs to be communicated are the $\mathbf{G}(t_k)$ matrices, which have a much smaller size than the full set of two-electron integrals.

An overall accuracy of 12 digits or more in the elements of the two-electron component of the Fock matrix can be obtained using a quadrature with the parameters $t_{\mathrm{lin}} = 4$, $t_{\mathrm{log}} = 10^5$, $N_{\mathrm{lin}} = 25$ and $N_{\mathrm{log}} = 25$, for a total $R = 51$ (including the tail point). This illustrated in Fig. 4.2 for some small systems, using the cc-pVDZ basis set [88].

The most expensive operation is the final multiplication of the $\mathbf{\Theta}$ tensors. The computational cost of this operation is proportional to the number of two electron integrals, $[(L+1)(L+2)/2]^4 \sim \mathcal{O}(L^8)$. This is the lowest possible scaling with respect of L, so for large basis sets this is potentially the most efficient algorithm developed so far.

Numerical calculation of three-electron integrals

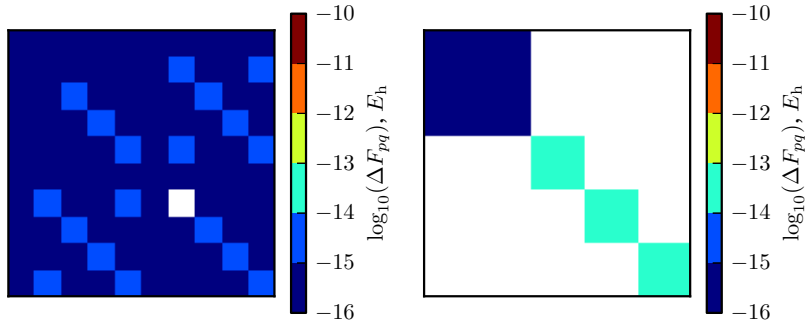
The scheme presented in the previous section can also be used to compute more complicated integrals, such as a type of three-electron integrals of interest in explicitly correlated methods:

$$G_{abcdef} = \iiint \chi_a^*(\mathbf{r}_1)\chi_b^*(\mathbf{r}_2)\chi_c^*(\mathbf{r}_3) r_{12}^{-1} r_{13}^{-1} \chi_d(\mathbf{r}_1)\chi_e(\mathbf{r}_2)\chi_f(\mathbf{r}_3) \mathrm{d}^3 r_1 \mathrm{d}^3 r_2 \mathrm{d}^3 r_3 \tag{4.46}$$

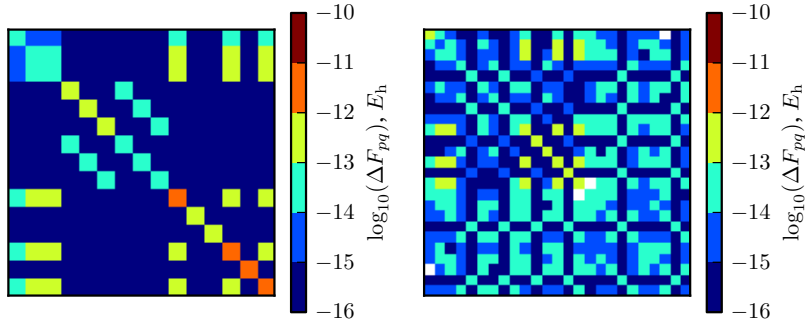
Analytic integration of (4.46) over GTOs has only been possible for the one-centre case [89].

By introducing the integral identity (3.24) for the operators r_{12}^{-1} and r_{13}^{-1} in the t and s dimensions, respectively, we obtain an expressions which resembles (4.34):

$$\begin{aligned} G_{abcdef} = & \frac{4}{\pi} K_{ad} K_{be} K_{cf} N^0_{abcdef} \\ & \times \int_0^\infty \int_0^\infty M_{PQS}(t,s) \Theta^x_{l_a^x l_b^x l_c^x l_d^x l_e^x l_f^x}(t,s) \Theta^y_{l_a^y l_b^y l_c^y l_d^y l_e^y l_f^y}(t,s) \Theta^z_{l_a^z l_b^z l_c^z l_d^z l_e^z l_f^z}(t,s) \mathrm{d}t\, \mathrm{d}s \end{aligned} \tag{4.47}$$



(a) H_2 (b) He



(c) Ne (d) H_2O

Figure 4.2 Error in the Fock matrix (in E_h) for some small closed-shell systems as obtained using the cc-pVDZ basis set. Each "pixel" represents the error in one matrix element. In the white areas, both the numerical and the analytical approaches yield identical elements of the Fock matrix.

with

$$M_{PQS}(t,s) = \frac{\pi^{9/2}\exp[-\Lambda(\alpha_Q,t)R_{PQ}^2 - \Lambda(\alpha_S,s)R_{PS}^2]}{\left[(\alpha_P+\Lambda(\alpha_Q,t)+\Lambda(\alpha_S,s))(\alpha_Q+t^2)(\alpha_S+s^2)\right]^{3/2}} \tag{4.48}$$

and

$$\Theta^{\xi}_{l_\xi^a l_\xi^b l_\xi^c l_\xi^d l_\xi^e l_\xi^f}(t,s) = \sum_{i=0}^{l_\xi^a+l_\xi^d}\sum_{j=0}^{l_\xi^b+l_\xi^e}\sum_{k=0}^{l_\xi^c+l_\xi^f} \tilde{T}^{P,\xi}_{l_\xi^a l_\xi^d i}\,\tilde{T}^{Q,\xi}_{l_\xi^b l_\xi^e j}\,\tilde{T}^{S,\xi}_{l_\xi^c l_\xi^f k}\,\Phi_{ijk}(t,s;\Xi_{PQ},\Xi_{PS}). \tag{4.49}$$

The two-dimensional polynomials $\Phi_{ijk}(\Xi_{PQ},\Xi_{PS})$ can be obtained using recursive rules

$$\Phi_{i+1,jk} = \frac{1}{2\alpha_P}\left[\left(-\partial_{\Xi_{PQ}} - \partial_{\Xi_{PS}}\right)\Phi_{ijk} + 2\left(\Lambda(\alpha_Q,t)\Xi_{PQ} + \Lambda(\alpha_S,s)\Xi_{PS}\right)\Phi_{ijk} + i\;\Phi_{i-1,jk}\right] \tag{4.50}$$

$$\Phi_{i,j+1,k} = \frac{1}{2\alpha_Q}\left[\partial_{\Xi_{PQ}}\Phi_{ijk} - 2\Lambda(\alpha_Q,t)\Xi_{PQ}\Phi_{ijk} + j\;\Phi_{i,j-1,k}\right] \tag{4.51}$$

$$\Phi_{ij,k+1} = \frac{1}{2\alpha_S}\left[\partial_{\Xi_{PS}}\Phi_{ijk} - 2\Lambda(\alpha_S,s)\Xi_{PS}\Phi_{ijk} + k\;\Phi_{ij,k-1}\right], \tag{4.52}$$

starting from $\Phi_{000} = 1$, using the shorthand notation $\partial_x \equiv \partial/\partial x$.

The integration over each coordinate is performed using the same quadrature as for the two-electron integrals. Similarly to the case of the two-electron integrals, in order to use the same quadrature for all integrals, numerical integration in the additional dimension is carried out in linearly transformed coordinates, namely

$$t' = \left(\sqrt{\frac{1}{\alpha_P}+\frac{1}{\alpha_Q}} + R_{PQ}\right)t \tag{4.53}$$

and

$$s' = \left(\sqrt{\frac{1}{\alpha_P}+\frac{1}{\alpha_S}} + R_{PS}\right)s. \tag{4.54}$$

4.3 Numerical real-space representations

The atom-centred basis sets introduced in the previous section, in particular the Gaussian-type orbitals, are by far the most common type of basis set used in quantum chemical calculations. Of the characteristics of the ideal basis set discussed in Section 4.1, atom-centred basis sets stand out because of their efficiency, as was earlier demonstrated, but also because of their compactness: the most complicated feature of the molecular orbitals, the cusps at the nuclear positions, can be rather faithfully represented with just a few Gaussian-type orbitals. A good example of this is how the tiny STO-3G basis set [84], consisting of three and fifteen Gaussian functions for first and second-row elements respectively, can qualitatively reproduce structures and energies of simple molecules at the Hartree–Fock level of theory. This is justified by the convergence with the maximum angular momentum number in the basis set L: the largest contribution is due to functions of low angular momentum number, with higher angular momentum number shells contributing less and less.

Concerning accuracy, GTOs typically perform adequately. Chemical accuracy, usually defined as errors of 1 kcal/mol $\sim 1.6\cdot 10^{-3}\ E_\mathrm{h}$ or less, can be obtained with not too large Gaussian basis sets. Furthermore, Gaussian basis sets largely benefit from cancellation of

errors. This is because Gaussian basis sets often yield similar errors for different systems, which are cancelled out when energy differences are computed.

The weakest point of atom-centred basis sets is that they are not systematic. In other words, there are no procedures to generate basis sets in which a variation of some parameters will certainly and indefinitely decrease the error of the calculation. For small basis sets, increasing the amount of functions will almost surely improve the results. However, as the amount of basis functions grows, basis sets become overcomplete. In other words, linear dependencies appear as some basis function can be faithfully represented as a linear combination of other basis functions. This causes numerical instabilities that in practice limit the minimum error attainable with atom-centred basis sets.

The second weak point of atom-centred bases is that they are not universal. This is due to the optimization required for their generation. They will work adequately for those circumstances for which they were fitted, but probably work poorly in others. The use of specialized basis sets, particularly for computing certain molecular properties, is necessary to obtain properly converged results.

There is one underlying reason for both the advantages and disadvantages of atom-centred basis functions: they have global support. The support of a function is the closure of the set of points in which they are non-zero. For atom-centred basis functions, the support is $\mathbb{R}^3$[†]. This means that when the coefficient of one basis function changes, *all* the other coefficients must change as well. This makes very difficult to indefinitely improve the quality of the representation.

The alternative is to use basis functions with compact support, that is, functions which are non-zero only inside a compact region of space. In such case, by using more basis functions with smaller support, it is possible to systematically improve the accuracy of the representation. Methods based on compact-support representations are commonly referred as grid-based methods and real-space methods. The reason is that the expansion coefficients are very often the values of the function at selected points of space: the grid points.

Grid-space methods can be considered orthogonal to atom-centred basis sets methods with respect to their advantages and disadvantages. First, they are systematic. Because they are not biased towards any system in particular, they are universal. However, the lack of bias implies that they do not benefit from cancellation of errors. Moreover, real-space methods require a large amount of grid points to provide an accuracy comparable to atom-centred basis sets. For the same reason, real-space methods are often much more computationally demanding than atom-centred basis set calculations.

In Fig. 4.3, the difference between Gaussian basis sets and real-space numerical representations is illustrated. Consider a Gaussian function $\exp\left(-x^2\right)$ in the interval $x \in [-2, 2]$. The function is approximated, on one hand, by fitting[‡] to an even-tempered basis [92], which is a GTO basis which can be made systematically more complete by means of some simple mathematical relations. On the other hand, it is approximated using a finite-element basis representation using an equidistant grid, as described later on p. 37. As it can be seen, the Gaussian basis does an excellent job for a small amount of coefficients, but then it is incapable of reducing the error further than 10^{-6}. Quite the opposite, the finite-element basis representation needs a fairly large amount of coefficients to yield the same accuracy, but then increasing the number of grid points systematically lowers the error. Note that in a more realistic three-dimensional calculation over a larger domain the amount of coefficients would be even larger.

Due to the large costs of real-space methods, applications for electronic structure calculations are uncommon. There are two special cases where they have seen application. The first is for systems where special coordinates can be used to treat the problem in one or two dimensions, such as atoms [93–100] and diatomic molecules [95–99, 101]. The second type of calculations where real-space methods have been successful comprises models

[†]While this is not true for numerical atomic orbitals, the cut-offs are large enough so that in practice they suffer from the same problems as is they had global support.

[‡]By minimizing the average quadratic error in the interval $[-2, 2]$ using a Simplex algorithm [90] as implemented in SciPy [91].

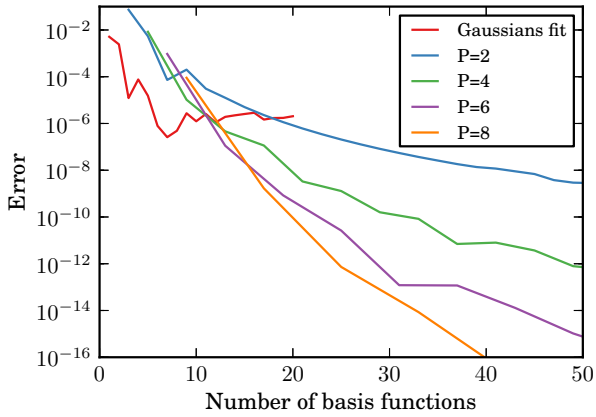


Figure 4.3 Average quadratic error for the representation of e^{-x^2} in the interval $[-2,2]$ using even-tempered Gaussian basis sets and finite-element bases on equidistant grids.

where the core electrons are taken care of in some implicit manner, such as pseudopotentials [102–104] or projector-augmented waves [105]. For all electron systems with arbitrary geometries applications have been fewer and restricted to systems containing a small amount of electrons [58,59,106], although remarkably large calculations can also be found in the literature [107].

The tensorial finite-element basis

A finite-element basis set consists of a set of functions $\{\chi_i\}$, each associated with a grid point $\{\mathbf{r}_i\}$, such that the function χ_i has support in a small region around $\mathbf{r}_i$. In this way, the overlap between most basis functions is 0.

A tensorial basis set is constructed as a tensor product of one-dimensional basis sets. For instance, a three-dimensional tensorial finite-element basis can be written as:

$$\left\{\chi_{ijk}(x,y,z)=\chi_i^x(x)\chi_j^y(y)\chi_k^z(z)\right\}=\left\{\chi_i^x(x)\right\}\otimes\left\{\chi_j^y(y)\right\}\otimes\left\{\chi_k^z(z)\right\} \tag{4.55}$$

The corresponding grid is a Cartesian product of the one-dimensional grids:

$$\{\mathbf{r}_{ijk}=(x_i,y_j,z_k)\}=X\times Y\times Z. \tag{4.56}$$

Three-dimensional functions are therefore expressed in the numerical basis as

$$f(x,y,z)=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z}f_{ijk}\chi_i^x(x)\chi_j^y(y)\chi_k^z(z). \tag{4.57}$$

The actual form of the finite-element basis can vary from one implementation to another. Piece-wise polynomials are a common choice. In the present work, the one-dimensional finite-element bases are constructed as follows.

The calculation domain $[x_{\min},x_{\max})\subset\mathbb{R}$ is divided into M right-open intervals, or *cells*, $\{A_i|1\leq i\leq M\}$. Each cell contains $P+1$ equidistant grid points, such that the last point of one cell is the first point of the next cell. The distance between the grid points of the i-th cell

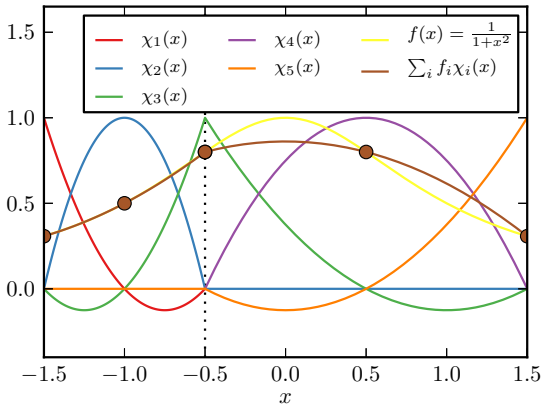


Figure 4.4 $f(x) = 1/(1+x^2)$ represented in the finite-element basis for $M = 2$ cells in the intervals $[-1.5, 0]$ and $[0, 1.5]$ using a LIP order of $P = 2$. The total number of basis functions is $N = 5$.

is called the *step*, and is denoted by h_i. The steps of all cells form the set $H = \{h_i | 1 \le i \le M\}$. The intervals occupied by the cells can then be written as $\{A_i = [s_i, s_i + Ph_i)\}$, where $s_i = x_{\min} + P\sum_{j=1}^{i-1} h_j$ is the starting point of the i-th cell. The grid is fully parametrized by $x_{\min}$, M, P and H. The total number of grid points is $N = MP + 1$.

For the construction of the basis set, we will use the *indicator function* $\mathbf{1}_A(x)$:

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases} \tag{4.58}$$

In other words, $\mathbf{1}_A(x)$ is 1 for all points belonging to the interval A, and 0 outside. In each cell, we construct a P-th order Lagrange interpolation polynomial basis (LIP), with the grid points of the cell as interpolation nodes. This is expressed as

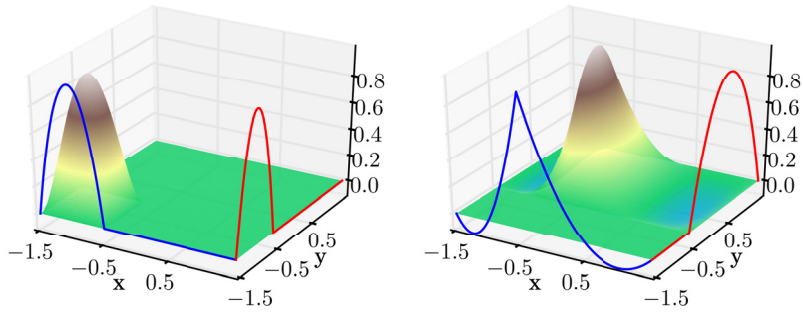
$$b_{ij}(x) = \mathbf{1}_{A_i}(x) \prod_{\substack{0 \le k \le P \\ k \ne j}} \frac{(x - s_i)/h_i - k}{j - k} \tag{4.59}$$

for $1 \le i \le M$ and $0 \le j \le P$. The basis set $\{\chi_i(x)\}$ is constructed from the functions $b_{ij}(x)$, with the functions at the junction points spanning two cells to ensure continuity:

$$\begin{aligned} \chi_1(x) &= b_{10}(x)\,; \\ \chi_{(i-1)P+j+1}(x) &= b_{ij}(x)\,; && 1 \le i \le M; 1 \le j \le P-1 \\ \chi_{(i-1)P+1}(x) &= b_{i-1,P}(x) + b_{i1}(x)\,; && 2 \le i \le M \\ \chi_N(x) &= b_{MP}(x). \end{aligned} \tag{4.60}$$

It should be noted that continuity in the derivatives is not enforced in any way. However, if the grids are sufficiently fine, the discontinuities can be made vanishingly small. The basis set can be trivially modified to include also the end point $x_{\max}$, by setting the point of the last basis function as $\chi_N(x_{\max}) = 1$.

An example basis set, for $M = 2$ and $P = 3$ is



(a) $\chi_2^x(x)\chi_2^y(y)$ (b) $\chi_3^x(x)\chi_4^y(y)$

Figure 4.5 Two two-dimensional basis functions. The one-dimensional grid in both dimensions is the same as in Fig. 4.4.

$$\begin{aligned}
&\{\chi_1(x) = b_{10}(x),\\
&\chi_2(x) = b_{11}(x),\\
&\chi_3(x) = b_{12}(x) + b_{20}(x),\\
&\chi_4(x) = b_{21}(x),\\
&\chi_5(x) = b_{22}(x)\}.
\end{aligned} \tag{4.61}$$

Since $\chi_i(x_j) = \delta_{ij}$, the expansion coefficients are the values of the interpolated function at the grid points,

$$f_i = f(x_i) \tag{4.62}$$

or, for a three-dimensional basis,

$$f_{ijk} = f(x_i, y_j, z_k). \tag{4.63}$$

A basis set in the range of $[-1.5, 1.5]$ with $M = 2$, $P = 2$ and steps $H = \{0.5, 1\}$ is illustrated in Fig. 4.4. The tensorial construction of the basis is illustrated in Fig. 4.5 in two dimensions. The one-dimensional grid in Fig. 4.4 is used in the x and y dimensions.

In the one-dimensional case, the function can be directly represented as a piece-wise sum of polynomials

$$f(x) = \sum_{i=0}^{M} \sum_{j=0}^{P} a_{i,j} \left(\frac{x - s_i}{h_i} \right)^j. \tag{4.64}$$

Nevertheless, the f_i values are stored instead. This simplifies the calculation of the expansion coefficients of new functions, such as $g(f(x)) = \sum_i g(f_i)\chi_i(x)$ or $g(x)f(x) = \sum_i f_i g_i \chi_i(x)$ by using inner projection. In the rest of this work, LIPs of order $P = 6$ are used.

One could argue that it would be more adequate to use more optimal interpolation points, such as the nodes of Chebyshev or Legendre polynomials. Although this would be true if one single polynomial were to be used for the whole range, it is not the case for the piece-wise polynomial basis employed here. This is illustrated in Fig. 4.6, where the differences in the errors are negligible for interpolating some representative function.

The main advantage of this numerical representation is the simplicity of the basis. The tensorial form of the basis allows a very efficient parallel implementation of several costly operations. However, due to the tensorial nature of the basis, unnecessary points must

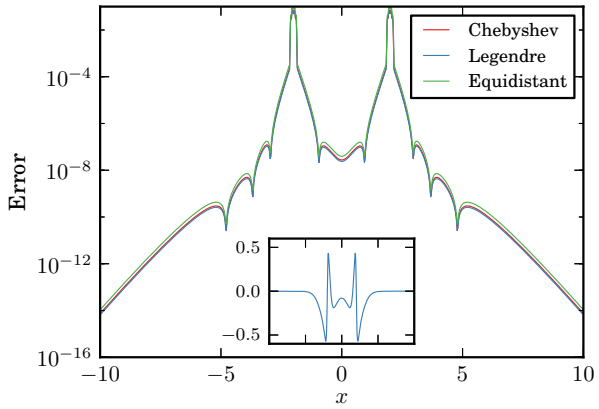


Figure 4.6 Error in the representation of a function using different interpolation nodes. The error is measured as $\frac{1}{D}\sqrt{\int_{x-D/2}^{x+D/2}|f(x')-P(x')|^2\mathrm{d}x'}$, $P(x)$ being the 6-th degree interpolating polynomial in the interval $[x-D/2, x+D/2]$, for $D=0.3$. The interpolated function, $f(x) = 8[(x+2)\mathrm{e}^{-6|x+2|}-(x-2)\mathrm{e}^{-6|x-2|}]-4[(x+2)^2\mathrm{e}^{-3|x+2|}+(x-2)^2\mathrm{e}^{-3|x-2|}]$, is represented in the inset.

be unavoidably stored, which is illustrated in Fig. 4.7. The problem with the amount of superfluous grid points can to some extent be mitigated by partitioning the domain into regions with similar grid requirements. However, a too fine subdivision complicates the basis and hampers an efficient parallelization of the methods described below.

The *bubbles* representation

As atom-centred basis sets excel where numerical grid representations fail, and vice versa, the question of designing a basis set containing the best of both worlds naturally arises. In other words, can the complicated nuclear cusps be described with atom-centred functions, while the smooth remainder is represented in some other way? Already the pseudopotential and projector-augmented wave methods described earlier realize these ideas, although in those cases the core orbitals are represented in an approximate manner. All-electron calculations have been performed using mixed basis sets such as combinations of Gaussians with plane waves [108] or finite elements [109]. Some early, fully numerical attempts for diatomic molecules were already carried out in the 1980s [110].

In the *bubbles* representation, a three-dimensional scalar function for a system with K atomic centres is partitioned as

$$f(\mathbf{r}) = \sum_{A=1}^{K} f^A(r_A, \theta_A, \phi_A) + f^\Delta(\mathbf{r}), \tag{4.65}$$

where the atom-centred functions $f^A(r_A, \theta_A, \phi_A)$ are the *bubbles* and the remainder $f^\Delta(\mathbf{r})$ is the *cube*. The bubbles consist of radial and angular parts:

$$f^A(r_A, \theta_A, \phi_A) = \sum_{l=0}^{L} \sum_{m=-l}^{l} f^{Alm}(r_A) Y_{lm}(\theta_A, \phi_A). \tag{4.66}$$

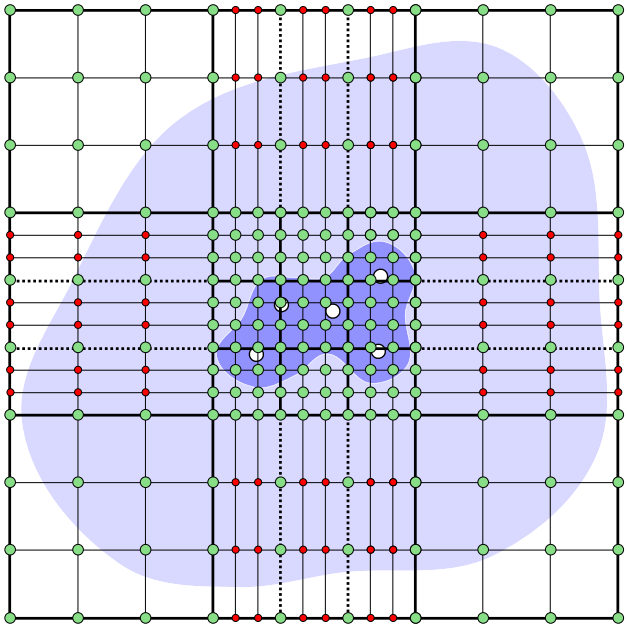


Figure 4.7 A two-dimensional cross-section of a non-equidistant grid with a fine inner region and a coarse outer region. An idealized function of a molecular system composed of 5 atoms (white circles), with a sharp, large-valued region (dark blue) and a smooth, small-value region (light blue) is represented. The red grid points are in practice superfluous, because these values must be stored, although they do not improve the overall accuracy of the calculation.

L is typically small, such as 1 or 2. The angular part of the bubbles are real spherical harmonics, $Y_{lm}(\theta,\phi)$, as in (4.8). The spherical coordinates r_A, θ_A and ϕ_A are relative to a nuclear position ($\mathbf{R}_A$).

The radial functions $f^{Alm}(r_A)$ and the cube $f^{\Delta}(\mathbf{r})$ are represented on one and three-dimensional tensorial finite-element bases respectively, that is

$$f^{Alm}(r_A)=\sum_i f_i^{Alm}\chi_i^A(r_A) \tag{4.67}$$

and

$$f^{\Delta}(\mathbf{r})=\sum_{ijk} f_{ijk}^{\Delta}\chi_i^x(x)\chi_j^y(y)\chi_k^z(z). \tag{4.68}$$

Throughout (4.65)–(4.68), the following notation has been used: for a function designated by a symbol f, the different functions that constitute it are denoted by superscripts (f^A, f^{Δ}, f^{Alm}), and the expansion coefficients into which those functions are expanded are given by subscripts (f_{ijk}^{Δ}, f_i^{Alm}).

The radial grid

The grid points in the radial interval $[0, r_{\max}]$ are obtained using an ad-hoc scheme that provides a fairly compact and accurate representation. However, the number of grid points can be increased without significantly affecting the overall performance, as most of the computational time is determined by the cube size. The coordinate of the beginning of cell i is obtained with the mapping

$$s_i=\frac{c_A r_{\max}(i-1)/M_A}{c_A+r_{\max}[1-(i-1)/M_A]}, \tag{4.69}$$

where $c_A=8Z_A^{-1.5}$, and Z_A is the charge of the nucleus where the A-th bubble is centred. To approximately preserve the same accuracy regardless of Z_A, the number of cells (M_A) is increased proportionally to $Z_A^{1/4}$. In the present calculations, $M_{Z=1}=200$ and $r_{\max}=20\ \mathrm{a_0}$ have been used.

To assess the accuracy of the radial grids, we construct radial electron densities of the form $2\sum_i \zeta_i^3/\pi \mathrm{e}^{-2\zeta_i r}$, mimicking noble gas atoms from He to Xe. The exponents are chosen from the largest exponents for each shell of the double-zeta all-electron STO basis of Van Lenthe and Baerends [111], summarized in Table 4.2. The relative errors in the electrostatic potentials computed using (3.14) are shown in Fig. 4.8. The radial grid provides at least 13 digit precision for elements in the first to third periods. The error worsens significantly for fourth and fifth row elements. For third row elements, the error is still tolerable, as at least 9 correct digits are obtained. However, for calculations involving heavier elements different procedures to generate radial grids should be explored.

The bubbles are represented by $(L+1)^2(M_AP+1)$ coefficients per atomic centre. For the parameters used here to build the radial grids and $L=2$ the number of coefficients per atom ranges between 5 and 15 thousand corresponding to 100 kB when using 64-bit floating numbers. As it will be shown below, this is moderate compared to the total storage costs for one function.

The radial functions can also include an analytical radial factor:

$$f^{Alm}(r_A)=g(r)\sum_{i=1}^{M} f_i^{Alm}\chi_i(r_A). \tag{4.70}$$

For instance, $g(r)=r^{-1}$ or $g(r)=r^{-2}$ can be used to circumvent singularities at the origin of the bubbles when representing functions like nuclear potentials or Laplacians, akin to atomic structure calculations, [93, 100].

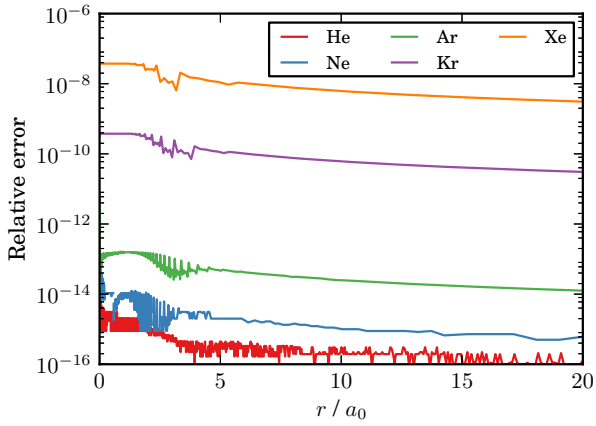


Figure 4.8 Relative error in the electrostatic potential for radial charge densities of the form $2\sum_i \zeta_i^3/\pi e^{-2\zeta_i r}$, where ζ_i are given in Table 4.2.

Table 4.2 Exponents used to generate the radial densities $2\sum_i \zeta_i^3/\pi e^{-2\zeta_i r}$ used in Fig. 4.8.

	$1s$	$2s$	$2p$	$3s$	$3p$	$3d$	$4s$	$4p$	$4d$	$5s$	$5p$
He	1.25										
Ne	12.45	3.65	4.00								
Ar	21.90	6.65	9.30	3.25	2.80						
Kr	43.60	14.35	19.85	7.35	8.15	9.05	3.40	2.90			
Xe	397.00	62.70	99.30	12.55	13.45	22.35	8.95	7.25	6.05	3.60	2.950

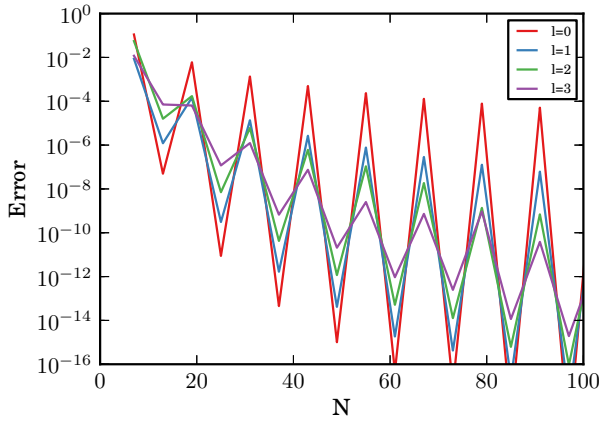


Figure 4.9 Average quadratic error in the representation of $e^{-|x|}x^l$ in the interval $[-5,5]$, as a function of the total number of grid points N. The LIP order is $P=6$.

The three-dimensional grid

The Cartesian grid for the cube must be constructed according to two criteria. First, its extent must be sufficient to contain most of the electron density. Second, the grid must be fine enough so that the considered functions can be accurately represented.

So far, a simple equidistant grid has been used. The cube grid ranges are chosen such that the domain boundaries are at least 8 a_0 apart from all atomic centres. The employed grid steps range from 0.05 to 0.4 a_0. For a typical grid step of 0.1 a_0, the total number of cube grid points per dimension is 100–200, for a total $N_{\text{cub}} = N_x N_y N_z$ of 10^6–10^7 coefficients (10–100 MB using 64-bit floating-point numbers). This is several orders of magnitude larger than the memory requirements for the bubbles.

Non-equidistant grids, as the one depicted in Fig. 4.7, can be used to reduce the amount of grid points. However, the savings become less significant as the size of the system grows.

Accuracy

As it was discussed earlier, increasing the number of grid points, both in the radial grids of the bubbles and in the cube, is expected to systematically lower the error in the representation. There is one caveat: as mentioned in Section 4.1, the derivatives of some of the represented functions are discontinuous at the nuclear position. For the charge density, this is the case already for the first derivatives. Unless the atomic centres are located exactly at the junction of eight cells, accurately representing such features with polynomials is very difficult. Fortunately, the bubbles-cube partition can be done in such a way that for a bubbles expansion with angular momentum number L, the derivatives of the cube at the nuclear positions are continuous up to order $L+1$. The higher the order of the first discontinuous derivative, the easier it becomes.

This is illustrated in Fig. 4.9 for a one-dimensional Slater function $e^{-|x|}x^l$ (with a $l+1$-th discontinuous derivative) in the interval $[-5,5]$, using equidistant elements with $P=6$. For an odd number of cells, the error converges very slowly, as the interpolating polynomial is not able to accurately represent the cusp. For an even number of cells, the cusp is located exactly at a cell junction, and the error becomes much smaller. This is the reason for the observed oscillations. However, as l increases the errors and the oscillations significantly decrease.

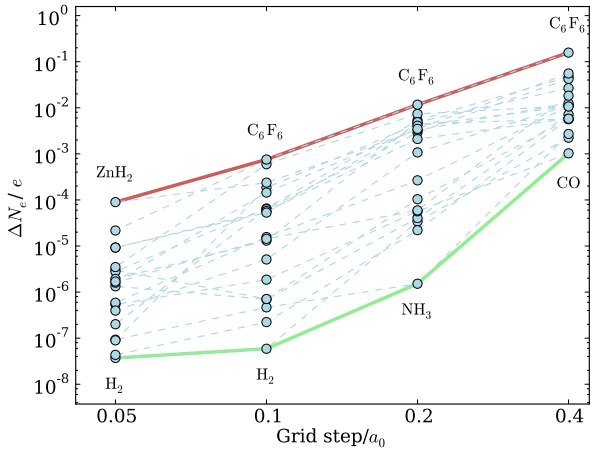


Figure 4.10 Error in the number of electrons ($N_e = \int \rho(\mathbf{r})\mathrm{d}^3 r$) with respect to the grid step. The trends for each individual molecule are represented with dashed lines. The wider lines represent the largest and smallest errors for each grid step. The molecule yielding the largest and smallest errors is also given for each grid step.

The accuracy of the representation is shown in Fig. 4.10, where the error in the number of electrons (i.e., the integral of the electron density over all space) is plotted as a function of the cube grid step for a variety of molecular systems. The details of the grid, the partitioning algorithm and the electronic structure used to compute the electron densities are discussed in Publication II. Although the accuracy varies largely for different molecules, the general trend is that the error decreases exponentially with the grid step, such that it is possible to obtain an error of less than 10^{-4} electrons for all systems.

Memory requirements

The memory requirements for the bubbles representation is largely dominated by the size of the cube. As the volume of the cube grows approximately linearly with the number of atoms as $aK + b$, so does the storage cost. This is illustrated in Fig. 4.11 for grids with a step of 0.1 a_0, for some molecular test systems (see Publication V). An "empty box" of roughly $16 \times 16 \times 16\ a_0^3$ would require $b = 4.40 \cdot 10^6$ coefficients (ca. 40 MB for 64 bit floats), and each atom added requires $a = 6.25 \cdot 10^5$ coefficients more (approximately 5 MB per atom). Halving the grid step increases the number of coefficients approximately by a factor of 8.

Addition and subtraction

The addition and subtraction of functions can be easily computed by calculating the values of the resulting function at the grid points. This procedure implies an inner projection of the resulting function into the basis set of the input function. Computing the sum or difference between two functions $h(\mathbf{r}) = f(\mathbf{r}) + g(\mathbf{r})$ can be done in a piecewise manner:

$$h^{\Delta}_{ijk} = f^{\Delta}_{ijk} + g^{\Delta}_{ijk} \tag{4.71}$$

$$h^{Alm}_{i} = f^{Alm}_{i} + g^{Alm}_{i}. \tag{4.72}$$

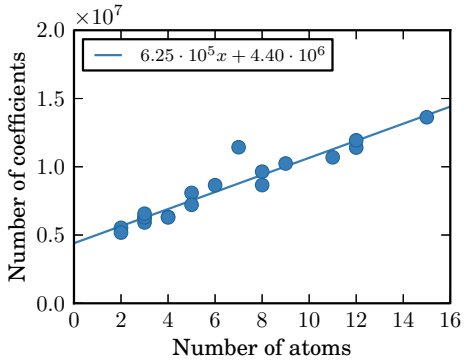


Figure 4.11 Number of coefficients required to store the electron density as a function of the number of atoms, for the equidistant grids with a step of 0.1 a_0. The outlier is H_2SO_4.

Integration

Integrations can be carried out efficiently and accurately in a piecewise manner. When integrating the *bubbles*, only the s contributions needs to be taken into account, because due to the symmetry of the spherical harmonics,

$$\int_0^{2\pi}\int_0^{\pi} Y_{lm}(\theta_A,\phi_A)\sin(\theta_A)\mathrm{d}\theta_A\mathrm{d}\phi_A = 4\pi\delta_{l0}. \tag{4.73}$$

Hence,

$$\int_{\mathbb{R}^3} f(\mathbf{r})\mathrm{d}^3r = 4\pi\sum_A\int_0^{\infty} f^{A00}(r_A)r_A^2\mathrm{d}r_A + \int_{\mathbb{R}^3} f^{\Delta}(\mathbf{r})\mathrm{d}^3r. \tag{4.74}$$

The integrals for the three-dimensional and one-dimensional parts are then given by

$$\int_0^{\infty} f^{A00}(r_A)r_A^2\mathrm{d}r_A = \sum_i f_i^{A00}r_i^2\left[\int_{-\infty}^{\infty}\chi_i(r)\mathrm{d}r\right] \tag{4.75}$$

and

$$\int_{\mathbb{R}^3} f^{\Delta}(\mathbf{r})\mathrm{d}^3r = \sum_{ijk} f_{ijk}^{\Delta}\left[\int_{-\infty}^{\infty}\chi_i^x(x)\mathrm{d}x\right]\left[\int_{-\infty}^{\infty}\chi_j^y(y)\mathrm{d}y\right]\left[\int_{-\infty}^{\infty}\chi_k^z(z)\mathrm{d}z\right]. \tag{4.76}$$

Equations (4.75) and (4.76) can be regarded as a dot product and a series of tensor-vector contractions, respectively.

Two-function products

Let us examine how to multiply two functions $f(\mathbf{r})g(\mathbf{r}) = h(\mathbf{r})$. In principle, because both $f(\mathbf{r})$ and $g(\mathbf{r})$ are accurately known at every point of space, they can be multiplied in a point-wise manner. However, a more desirable approach is to obtain the resulting function $h(\mathbf{r})$ directly in the bubbles representation. This cannot be done in a straightforward manner, as complicated cross-terms appear:

$$
\begin{aligned}
h(\mathbf{r}) = f(\mathbf{r})g(\mathbf{r}) = &\sum_A f^A(r_A,\theta_A,\phi_A)g^A(r_A,\theta_A,\phi_A) + \\
&\sum_{B\neq A} f^A(r_A,\theta_A,\phi_A)g^B(r_B,\theta_B,\phi_B) + \\
&\sum_A f^A(r_A,\theta_A,\phi_A)g^\Delta(\mathbf{r}) + \\
&\sum_A f^\Delta(\mathbf{r})g^A(r_A,\theta_A,\phi_A) + \\
&f^\Delta(\mathbf{r})g^\Delta(\mathbf{r}).
\end{aligned} \tag{4.77}
$$

Only the first and last term preserve the properties of the *bubbles* and *cube*, respectively.

The product in (4.77) can be expanded as

$$
\Big[\sum_A f^A(r_A,\theta_A,\phi_A) + f^\Delta(\mathbf{r})\Big]\Big[\sum_A g^A(r_A,\theta_A,\phi_A) + g^\Delta(\mathbf{r})\Big] = \sum_A h^A(r_A,\theta_A,\phi_A) + h^\Delta(\mathbf{r}). \tag{4.78}
$$

The following expression can be used to directly compute the radial parts of of $h(\mathbf{r})$:

$$
\begin{aligned}
&h^{Alm}(r_A) = \\
&\sum_{l_1m_1}\sum_{l_2m_2}\left\langle Y_{lm}|Y_{l_1m_1}Y_{l_2m_2}\right\rangle\left[f^{Al_1m_1}(r_A)\tilde{g}^{Al_2m_2}(r_A)\mu(r_A) + \tilde{f}^{Al_1m_1}(r_A)g^{Al_2m_2}(r_A)\nu(r_A)\right]
\end{aligned} \tag{4.79}
$$

where $\left\langle Y_{lm}|Y_{l_1m_1}Y_{l_2m_2}\right\rangle$ is a Clebsch–Gordan coefficient. The functions $\mu(r)$ and $\nu(r)$ ensure the correct long range behavior, and depend on the properties of $f(\mathbf{r})$ and $g(\mathbf{r})$. For a function $f(\mathbf{r})$, the radial functions $\tilde{f}^{Alm}(r)$ are given by

$$
\tilde{f}^{Alm}(r_A) = f^{Alm}(r_A) + \sum_{t=0}^{T} r_A^t \sum_{\{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=t\}} D_{lm}^{\boldsymbol{\alpha}} \frac{(\partial_{\mathbf{r}}^{\boldsymbol{\alpha}} f^{\neq A})(\mathbf{R}_A)}{\boldsymbol{\alpha}!}. \tag{4.80}
$$

The function $f^{\neq A}(\mathbf{r}$ appearing in (4.80), which denotes all parts of $f(\mathbf{r})$ except for the bubble centred at atom A,

$$
f^{\neq A}(\mathbf{r}) = f(\mathbf{r}) - f^A(\mathbf{r}) = \sum_{B\neq A} f^B(\mathbf{r}) + f^\Delta(\mathbf{r}), \tag{4.81}
$$

is expanded in a truncated Taylor series of order T. $D_{lm}^{\boldsymbol{\alpha}}$ is the projection of Y_{lm} on the Cartesian basis vector $\mathbf{r}_A^{\boldsymbol{\alpha}} = x_A^{\alpha_1} y_A^{\alpha_2} z_A^{\alpha_3}$:

$$
D_{lm}^{\boldsymbol{\alpha}} = \int_0^{2\pi}\int_0^{\pi} \hat{\mathbf{r}}_A^{\boldsymbol{\alpha}} Y_{lm}(\theta_A,\phi_A)\sin(\theta_A)\mathrm{d}\phi_A\mathrm{d}\theta_A. \tag{4.82}
$$

Obtaining $\tilde{f}_{lm}^A(r_A)$ is relatively simple, requiring the evaluation of Cartesian derivatives of $f^\Delta(\mathbf{r})$ and all $f^A(\mathbf{r})$ at the centre of every *bubble*.

The rationale behind (4.80) is that $\tilde{f}_{lm}^A(r_A)$ are good approximations to $f(\mathbf{r})$ in the vicinity of $\mathbf{R}_A$, using only the bubbles expansion. As the order of the Taylor series, T, grows

$$
\lim_{T\to\infty}\sum_{lm}\tilde{f}^{Alm}Y_{lm}(\theta_A,\phi_A) = f(\mathbf{r}) \tag{4.83}
$$

and hence the remainder $h^\Delta(\mathbf{r})$ decreases.

For a truncated Taylor series of order T, $\sum_{lm} \tilde{f}^{Alm} Y^A_{lm}$ is at long distances proportional to r_A^T, and hence diverges. Therefore, so will the product $\left(\sum_{lm} \tilde{f}^{Alm} Y^A_{lm}\right) g^A(\mathbf{r})$, unless $g^A(\mathbf{r})$ decays faster. For instance, for a product of two orbitals, which decay like e^{-r}, this is not a problem, but for a product or an orbital and a potential decaying like r^{-1} it is. The functions $\mu(r)$ and $\nu(r)$ impose the correct long-range behaviour. For instance, for the case of multiplying two orbitals, $\mu(r) = \nu(r) = 1$ works fine. For a product of an exponentially decaying function and a potential caused by a continuous charge distribution, which is more or less smooth, the second term can be neglected by setting $\nu(r)$ to 0.

The result of the bubbles multiplication procedure is illustrated in Fig. 4.12, for the product $U(\mathbf{r}) = \rho(\mathbf{r})V(\mathbf{r})$, where the charge density $\rho(\mathbf{r})$ is the superposition of two hydrogen 1s charge densities (Eq. (4.10)) centred at $(1,0,0)$ and $(-1,0,0)$, and $V(\mathbf{r})$ is the potential caused by $\rho(\mathbf{r})$.

Once the radial functions $h^{Alm}(r)$ have been computed, $h^\Delta(\mathbf{r})$ can be then calculated as a remainder:

$$h^\Delta_{ijk} = f(\mathbf{r}_{ijk})g(\mathbf{r}_{ijk}) - \sum_A h^A(\mathbf{r}_{ijk}). \tag{4.84}$$

Computing the cube of the resulting function is the costliest operation, because all radial functions and spherical harmonics must be evaluated at every *cube* grid point. The algorithm can be written down as follows:

for all A bubbles **do**
  • *Precompute interpolating polynomials*
  for all $l = 0, \ldots, L$ **do**
    for all $m = -l, \ldots, l$ **do**
      for all $i = 1, \ldots, M_A$ **do**
        Compute a_{ij}^{Alm} such that
        $w_i(x)\sum_j a_{ij}^{Alm}((x - s_i)/h_i)^j = \sum_j f_i^{Alm} b_{ij}(x)$
  • *Interpolate the bubbles*
  for all $\mathbf{r}_{\alpha\beta\gamma} = (x_\alpha, y_\beta, z_\gamma)$ grid points **do**
    • *Compute relative coordinates, distance, unit vector*
    $\mathbf{r}_A \leftarrow \mathbf{r}_{\alpha\beta\gamma} - \mathbf{R}_A$
    $r_A \leftarrow |\mathbf{r}_A|$
    $\hat{\mathbf{r}}_A \leftarrow \mathbf{r}_A / r_A$
    • *Find cell*
    Find i such that $s_i \le r_A \le s_i + P h_i$
    • *Compute local coordinate*
    $q \leftarrow (r_A - s_i)/h_i$
    for all $1 \le l \le L, -l \le m \le l$ **do**
      • *Evaluate interpolating polynomial*
      $f \leftarrow \sum_{0 \le j \le P} a_{ij}^{Alm} q^j$
      • *Evaluate spherical harmonic*
      for all (u, v, w) such that $C^{lm}_{uvw} \neq 0$ **do**
        $Y \leftarrow Y + C^{lm}_{uvw} \hat{x}_A^u \hat{y}_A^v \hat{z}_A^w$
      • *Accumulate result*
      $f_{\alpha\beta\gamma} \leftarrow f_{\alpha\beta\gamma} + fY$

First, the coefficients of the interpolating polynomials are computed, as of (4.64). The computational cost for this is negligible. Then, for every grid point, the bubble is interpolated. This consists of a series of steps, namely computing the relative coordinates of the grid point with respect to the bubble centre, find the cell in the radial grid, transform to cell coordinates, and, for every l and m, the radial function is interpolated and the spherical

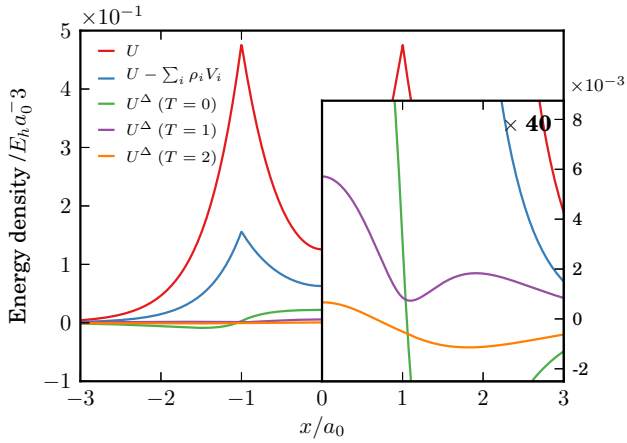


Figure 4.12 U^{Δ} along the x axis for a superposition of two $1s$ charge densities of hydrogen-like atoms with $Z = 1$ at $x = \pm 1\ a_0$, after subtracting bubbles generated with different Taylor series orders T. The result of subtracting the "diagonal elements", $U - \sum_i \rho_i V_i$, is shown for comparidon. The right hand side of the graph is magnified to show $U^{\Delta}(T = 2)$.

harmonic evaluated. The computational cost is the approximately $NK[a + b(l_{max} + 1)^2]$, where a is the cost of computing the relative coordinates and finding the cell in the radial grid, and b is the cost of evaluating one interpolating polynomial and one spherical harmonic.

Linear transformations

Linear transformations $g(\mathbf{r}) = \hat{O}f(\mathbf{r})$ are conveniently dealt with, as they can be applied in a piece-wise manner:

$$\hat{O}f(\mathbf{r}) = \sum_{A}^{K} \hat{O}f^{A}(r_A, \theta_A, \phi_A) + \hat{O}f^{\Delta}(\mathbf{r}). \tag{4.85}$$

Many relevant operators encountered in electronic structure calculations are linear. Some examples are the Laplacian operator $\nabla^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$ and the convolution with the Poisson and Helmholtz kernel discussed earlier.

If the operator can be written in a simple way in spherical coordinates, its effect on the bubbles can be computed fast and accurately. The effect on the cube can be computed in an efficient manner if the operator can be rewritten in a Cartesian-separated form, in order to exploit the tensorial nature of the basis:

$$\hat{O} \approx \sum_{p} \omega_p \hat{O}_x^p \hat{O}_y^p \hat{O}_z^p. \tag{4.86}$$

In such case, the coefficients of the resulting function can be obtained as

$$(\hat{O}f^{\Delta}(\mathbf{r}))_{ijk} \approx \sum_{p}^{R} \omega_p \sum_{k'}^{N_z} O_{kk'}^{z,p} \sum_{j'}^{N_y} O_{jj'}^{y,p} \sum_{i'}^{N_x} O_{ii'}^{x,p} f_{i'j'k'}^{\Delta} + C f_{ijk}^{\Delta} \tag{4.87}$$

The elements of the $\mathbf{O}^p$ matrices are

$$O_{ii'}^{\xi,p} = \hat{O}^{\xi,p}\chi_{i'}(\xi)\Big|_{\xi=\xi_i} \tag{4.88}$$

The number R of $\mathbf{O}$ matrices, the explicit form of their elements, and the values of the coefficients $\{\omega_p\}$ and C depend on the operator in question.

The expression in (4.87) can be recast as a tandem of matrix multiplications. Here, we follow the tensor notation of Kolda et al. [112]. The three-index tensor containing the elements of the cube of the input function f_{ijk}^{Δ} is denoted as $\mathbf{F}$. Similarly, $\mathbf{V}$ denotes the cube tensor of the output function. $\mathbf{G}$ and $\mathbf{H}$ are two- and three-dimensional intermediate tensors. Two-index slices are expressed as e.g. $\mathbf{F}_{:i:}$, which is a matrix whose elements are $(\mathbf{F}_{:i:})_{jk} = (\mathbf{F})_{jik}$. The operator matrices are stored as three-dimensional tensors $\mathbf{O}^x$, $\mathbf{O}^y$ and $\mathbf{O}^z$, with $(\mathbf{O}^{\xi})_{ii'p} = O_{ii'}^{\xi,p}$. The algorithm is outlined as

for all $1 \leq p \leq R$ **do**
  for all k' slices along the z axis **do**
    $\mathbf{G} \leftarrow \mathbf{O}^x_{::p}\mathbf{F}_{::k'}$
    $\mathbf{H}_{::k'} \leftarrow \mathbf{G}(\mathbf{O}^y_{::p})^{\mathbf{T}}$
  for all j slices along the y axis **do**
    $\mathbf{V}_{:j:} \leftarrow \mathbf{V}_{:j:} + \omega_p \mathbf{H}_{:j:}(\mathbf{O}^z_{::p})^{\mathbf{T}}$
$\mathbf{V} \leftarrow \mathbf{V} + C\mathbf{F}$

The matrix multiplications are carried out in the order x, y then z, because the elements of the cube and the operator matrices are stored in column-major order. All in all, $R(N_x + N_y + N_z)$ matrix multiplications are performed, with a total floating-point operation count of $2RN_xN_yN_z(N_x + N_y + N_z)$. For a cubic grid with $N = N_x = N_y = N_z$ this amounts to approximately $6RN^4$. Because of the linear increase of N_{cub} with the number of atoms K, the computational cost is expected to grow proportionally to $R(aK + b)^{4/3}$.

Identity/projection operator The identity operator,

$$f(\mathbf{r}) = \hat{I}f(\mathbf{r}) \tag{4.89}$$

is the most simple form of linear operator to be considered. It can be actually used to project a function onto a different grid. The operation on the bubbles would be simply a copy, and the cube operator would have rank $R = 1$, with matrix elements given by

$$O_{ii'}^{\xi,1} = \chi_{i'}^{\xi}(\xi_i) \tag{4.90}$$

The Laplacian operator The spherical harmonics are the eigenfunctions of the Laplacian operator. Hence, it is only necessary to compute the resulting radial functions, given by

$$\nabla^2 f^{Alm}(r)Y_{lm}(\theta,\phi) = \frac{1}{r^2}\left[\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}}{\mathrm{d}r}f^{Alm}(r)\right) - l(l+1)f^{Alm}(r)\right]Y_{lm}(\theta,\phi) \tag{4.91}$$

The r^{-2} factor can be stored implicitly, instead of explicitly on a radial grid, to avoid numerical problems.

For the cube, the Laplacian can be exactly represented as an operator of rank $R = 3$, with the coefficients appearing in (4.87) set to $\omega_1 = \omega_2 = \omega_3 = 1$, $C = 0$, and the operator matrix elements given by:

Table 4.3 Some examples of periodic systems that can be constructed using (4.96).

K_x	K_y	K_z	System
$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	Crystal
$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}^*$	Surface
$\mathbb{Z}$	$\mathbb{Z}$	$\{0\}$	Infinite slab
$\mathbb{Z}$	$\{0\}$	$\{0\}$	Wire
$\mathbb{Z}^*$	$\{0\}$	$\{0\}$	Semi-infinite wire
$\{-1,0,1\}$	$\{-1,0,1\}$	$\{-1,0,1\}^*$	Cube of $3\times3\times3$ cells

$$O^{x,1}_{ii'} = \partial^2_x \chi^x_{i'}(x_i); \quad O^{y,1}_{jj'} = \chi^y_{j'}(y_j); \quad O^{z,1}_{kk'} = \chi^z_{k'}(z_k) \tag{4.92}$$

$$O^{x,2}_{ii'} = \chi^x_{i'}(x_i); \quad O^{y,2}_{jj'} = \partial^2_y \chi^y_{j'}(y_j); \quad O^{z,2}_{kk'} = \chi^z_{k'}(z_k) \tag{4.93}$$

$$O^{x,3}_{ii'} = \chi^x_{i'}(x_i); \quad O^{y,3}_{jj'} = \chi^y_{j'}(y_j); \quad O^{z,3}_{kk'} = \partial^2_z \chi^z_{k'}(z_k) \tag{4.94}$$

Note again that, because the basis has discontinuous derivatives at the cell edges, the accuracy of this approach depends on the smoothness of the cube.

Convolution with the Coulomb and Poisson kernels One of the main purposes of the bubbles representation is to accurately and efficiently calculate electrostatic potentials. For the bubbles, the radial functions of the potential can be computed using (3.14). The integrals can be very easily computed in the one-dimensional finite-element basis.

For the cube, the tensorial basis set is particularly suitable for the quadrature derived in Section 3.3, leading to an accurate, efficient and easily parallelizable method [42, 113]. For molecular systems, in Publication II it is shown how a precision of 6 to 8 digits in the electrostatic potential can be obtained using an operator rank of $R = 20$. The elements of the operator matrices are given by

$$O^{\xi,p}_{ii'} = \int_{-\infty}^{\infty} \mathrm{e}^{-t_p^2(\xi-\xi_i)^2} \chi_{i'}(\xi)\mathrm{d}\xi \tag{4.95}$$

As it is shown in Publication I, this approach is convenient for the treatment of a certain class of periodic systems. Given the lattice vector $\mathbf{A} = (a_x, a_y, a_z)$, and the translational indices $K_\xi \subset \mathbb{Z}$ for $\xi = x, y, z$, consider periodic systems that can be constructed as

$$\rho^\Delta(x,y,z) = \sum_{k_x\in K_x}\sum_{k_y\in K_y}\sum_{k_z\in K_z} \rho^\Delta_0(x-k_x a_x, y-k_y a_y, z-k_z a_z) \tag{4.96}$$

The system consists of charge densities $\rho^\Delta_0(\mathbf{r})$ laid out in an array given by the Cartesian product of the translational indices, $K_x \times K_y \times K_z$. Depending on the translational indices, different types of periodic systems can be constructed, as shown in Table 4.3.

The operator matrix elements for the convolution with the Poisson kernel are then obtained as

$$O^{\xi,p}_{ii'} = \sum_{k_\xi\in K_\xi} \int_{-\infty}^{\infty} \mathrm{e}^{-t_p^2(\xi-\xi_i-k_\xi a_\xi)^2} \chi_{i'}(\xi)\mathrm{d}\xi \tag{4.97}$$

Because the absolute value of the terms decays roughly as $\mathrm{e}^{-|k_\xi|^2}$, the series is convergent. It is nevertheless convenient to carry out the summation in increasing order of $|k_\xi|$,

i.e. 0, 1, -1, 2, -2, etc. For small values of t_p, converge is slower and the integral can require a large number of terms to converge accurately. In fully periodic three-dimensional systems, computing the electrostatic potential via direct summation is conditionally convergent. This problem emerges in the present approach as a difficulty to obtain accurately the tiny contributions from small t_p values. Therefore, this issue can be circumvented by starting the integration in t at some $t_i > 0$. This is discussed in detail in Publication I.

Implementation on GPGPUs

Traditionally, computing has been performed on central processing units (CPU). The increase of the speed of computers during the last decades has relied on Moore's law: roughly, the speed of processors is expected to double every two years. However, due to limits in the miniaturization of transistors, the trend is expected to eventually break down. The alternative route is parallelization, that is, running several tasks simultaneously in several computing cores.

The obvious approach is running the program on several CPUs running in parallel. Indeed, nowadays it is difficult to find a CPU which does not contain at least two or four computing cores. Besides CPUs, other parallel computer architectures exist. In the last years, general-purpose graphics processing units (GPGPUs) have emerged as an attractive platform for high performance computing. GPGPUs are single-instruction multiple-data (SIMD) devices, that is, they execute the same program instruction on different sets of data. In contrast to CPUs, GPGPUs comprise hundreds of computing cores of lower speed, with only a small part of the chip being used for other functions such as caching. GPGPUs have therefore higher theoretical peak performances than CPUs, often by one or two orders of magnitude. Also, GPGPUs have a much lower cost per FLOP, both in terms of electrical power and money.

However, in order to implement a given method in a parallel architecture, it first needs to be broken down into operations that can be run in parallel. Not every algorithm can be parallelized equally easily. Often, parallelization of an efficient serial algorithm leads to more expensive individual steps. In this case, only when the overheads derived from parallelization are sufficiently small is the parallel version faster.

There are two main things to consider when devising an algorithm for GPGPUs. The first is considering the SIMD nature of the device: ideally, all operations performed should be identical, for instance the algorithm should contain no branching, such as *if–else* blocks. The second is that the transfer of information between main memory to the graphics card memory is slow, and should therefore be minimized.

GPGPUs are very appealing for quantum chemical calculations. Unfortunately, although remarkable advances have been made in the recent years [81, 114], the most common algorithms are difficult to port to GPGPUs. As it has been shown in this chapter, the nature of the bubbles basis leads to algorithms that can be trivially parallelized. The operations are broken down into identical instruction blocks performed on different sets of input data. The most expensive operations using the bubbles basis are the transformations and multiplications of two functions, due to the bubbles injection step. In Publication V, the performance of these two steps has been benchmarked on GPGPUs. Both operations show remarkable speed. Despite the considerable cost of the presented methods, their accuracy and their suitability for massively parallel computer architectures makes the bubbles basis an attractive candidate for performing quantum chemical calculations.

5 Results and conclusions

In this thesis, a number of numerical methods applicable to calculations of electronic structure have been presented. A special focus has been devoted on producing highly parallelizable algorithms, suitable for the new generations of computers. Another point shared by the presented methods is the rigorous control of the numerical accuracy by some simple parameters.

The presented methods have been published in five articles, which are summarized in the following.

5.1 Summary of the papers

Paper I

In this work, the Direct Approach to Gravitation and Electrostatics (DAGE) [42, 113] was extended to treat a variety of periodic systems, such as surfaces and wires. The quadrature used in those previous articles proved to be insufficient, as it did not converge adequately for extended systems in three dimensions. This led to the development of the quadrature for the Coulomb potential presented in Section 3.3, used throughout the rest of the papers included in this thesis.

Sergio Losilla developed the numerical quadrature. He wrote the computer implementation, planned and ran the test calculations, and wrote the first draft of the manuscript.

Paper II

The *bubbles* framework for representing molecular scalar functions was developed. It was shown that an accurate and economic representation could be achieved by using a mixture of atom-centred bases and three-dimensional finite elements.

The numerical representation was originally conceived by Dage Sundholm, using functions of s and p symmetry to represent the nuclear cusps. Sergio Losilla extended it to spherical harmonics of arbitrary order. Sergio Losilla also developed the algorithm for computing products of two functions, implemented the computer library, devised the density partitioning method, and wrote the first version of the manuscript.

Paper III

In this paper, the quadrature for the Coulomb potential was applied to computing two-electron integrals over Gaussian-type orbitals. Recursive expressions for computing integrals over functions with arbitrary angular momentum number were derived. The obtained expressions were similar to those in the Obara–Saika method, but the algorithm has lower asymptotic scaling. The numerical quadrature breaks the problem into smaller pieces, opening new parallelization possibilities.

The mathematical formulation was derived by Mooses Mehine. Sergio Losilla assisted in deriving the equations, adapted the numerical quadrature for the purpose, wrote the computer program, computed the results and prepared the figures and tables, and wrote parts of the manuscript.

Paper IV

Analogously to Paper III, the quadrature developed in Paper I was applied to a class of three-electron integrals appearing in explicitly correlated calculations, which in general cannot be computed analytically.

Mooses Mehine derived the equations. Sergio Losilla extended the quadrature scheme to compute the two-dimensional integrals, implemented the computer program, carried out the calculations and wrote parts of the manuscript.

Paper V

The algorithms presented in Paper II were ported to run on GPGPUs. The algorithms showed excellent performance and suitability for these emerging many-core architectures. Sergio Losilla implemented the programs, performed the calculations, and wrote the bulk of the manuscript.

Bibliography

[1] K. Gavroglu and A. Simões. *Neither Physics nor Chemistry: A History of Quantum Chemistry*, (MIT Press, Cambridge, MA, 2012).

[2] P. A. M. Dirac. Quantum Mechanics of Many-Electron Systems. *Proc. R. Soc. A* **123**, 714 (1929).

[3] F. Jensen. *Introduction to Computational Chemistry*, (Wiley, Chichester, 1999).

[4] J. Olsen and P. Jørgensen. Linear and nonlinear response functions for an exact state and for an MCSCF state. *J. Chem. Phys.* **82**, 3235 (1985).

[5] M. Born and R. Oppenheimer. Zur Quantentheorie der Molekeln. *Ann. Phys.* **84**, 457 (1927).

[6] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science* **332**, 60 (2011).

[7] E. A. Hylleraas. Über den Grundzustand des Heliumatoms. *Zeits. Phys.* **48**, 469 (1928).

[8] E. A. Hylleraas. Neue Berechnung der Energie des Heliums im Grundzustande, sowie des tiefsten Terms von Ortho-Helium. *Zeits. Phys.* **54**, 347 (1929).

[9] E. A. Hylleraas. The Schrödinger Two-Electron Atomic Problem. *Adv. Quantum Chem.* **1**, 1 (1964).

[10] H. M. James and A. S. Coolidge. The ground state of the hydrogen molecule. *J. Chem. Phys.* **1**, 825 (1933).

[11] S. Larsson. Calculations on the ^2S Ground State of the Lithium Atom Using Wave Functions of Hylleraas Type. *Phys. Rev.* **169**, 49 (1968).

[12] J. S. Sims and S. Hagstrom. Combined Configuration-Interaction—Hylleraas-Type Wave-Function Study of the Ground State of the Beryllium Atom. *Phys. Rev. A* **4**, 908 (1971).

[13] W. Kutzelnigg. r_{12}-Dependent terms in the wave function as closed sums of partial wave amplitudes for large l. *Theor. Chim. Acta* **68**, 445 (1985).

[14] W. Kutzelnigg and W. Klopper. Wave functions with terms linear in the interelectronic coordinates to take care of the correlation cusp. I. General theory. *J. Chem. Phys.* **94**, 1985 (1991).

[15] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*, (Dover, New York, 1996).

[16] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev.* **136**, B864 (1964).

[17] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **140**, A1133 (1965).

[18] P. Boerrigter, G. te Velde and E. Baerends. Three-dimensional numerical integration for electronic structure calculations. *Int. J. Quant. Chem.* **33**, 87 (1988).

[19] G. te Velde and E. J. Baerends. Numerical integration for polyatomic systems. *J. Comp. Phys.* **99**, 84 (1992).

[20] O. Treutler and R. Ahlrichs. Efficient molecular numerical integration schemes. *J. Chem. Phys.* **102**, 346 (1995).

[21] K. Burke. Perspective on density functional theory. *J. Chem. Phys.* **136**, 150901 (2012).

[22] T. D. Crawford and H. F. Schaefer III. An introduction to coupled cluster theory for computational chemists. *Rev. Comp. Chem.* **14**, 33 (2000).

[23] C. Møller and M. S. Plesset. Note on an Approximation Treatment for Many-Electron Systems. *Phys. Rev.* **46**, 618 (1934).

[24] K. Raghavachari, G. W. Trucks, J. A. Pople and M. Head-Gordon. A fifth-order perturbation comparison of electron correlation theories. *Chem. Phys. Letters* **157**, 479 (1989).

[25] K. Andersson, P. Å. Malmqvist and B. O. Roos. Second-order perturbation theory with a complete active space self-consistent field reference function. *J. Chem. Phys.* **96**, 1218 (1992).

[26] J. C. Maxwell. A Dynamical Theory of the Electromagnetic Field. *Phil. Trans. R. Soc.* **155**, 459 (1865).

[27] G. B. Arfken and H. J. Weber. *Mathematical Methods for Physicists*, (Academic Press, San Diego, 2001), 5th edition.

[28] T. Helgaker, P. Jørgensen and J. Olsen. *Molecular Electronic-Structure Theory*, (Wiley, Chichester, 2000).

[29] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.* **73**, 325 (1987).

[30] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon. The Continuous fast multipole method. *Chem. Phys. Letters* **230**, 8 (1994).

[31] M. C. Strain, G. E. Scuseria and M. J. Frisch. Achieving linear scaling for the electronic quantum Coulomb problem. *Science* **271**, 51 (1996).

[32] M. A. Watson, P. Sałek, P. Macak and T. Helgaker. Linear-scaling formation of Kohn-Sham Hamiltonian: application to the calculation of excitation energies and polarizabilities of large molecular systems. *J. Chem. Phys.* **121**, 2915 (2004).

[33] J. R. Chelikowsky, N. Troullier, K. Wu and Y. Saad. Higher-order finite-difference pseudopotential method: An application to diatomic molecules. *Phys. Rev. B* **50**, 11355 (1994).

[34] T. L. Beck. Real-space mesh techniques in density-functional theory. *Rev. Mod. Phys.* **72**, 1041 (2000).

[35] R. J. F. Berger and D. Sundholm. A non-iterative numerical solver of Poisson and Helmholtz equations using high-order finite-element functions. *Adv. Quantum Chem.* **50**, 235 (2005).

[36] A. Castro, A. Rubio and M. J. Stott. Solution of Poisson's equation for finite systems using plane-wave methods. *Can. J. Phys.* **81**, 1151 (2003).

[37] A. A. Coelho and R. W. Cheary. A fast and simple method for calculating electrostatic potentials. *Comput. Phys. Commun.* **104**, 15 (1997).

[38] G. Fann, G. Beylkin, R. J. Harrison and K. E. Jordan. Singular operators in multiwavelet bases. *IBM J. Res. Dev.* **48**, 161 (2004).

[39] S. Goedecker and O. V. Ivanov. Linear scaling solution of the Coulomb problem using wavelets. *Solid State Commun.* **105**, 665 (1998).

[40] L. Plagne and J. Y. Berthou. Tensorial Basis Spline Collocation Method for Poisson's Equation. *J. Comput. Physics* **157**, 419 (2000).

[41] A. Stathopoulos, S. Ogut, Y. Saad, J. R. Chelikowsky and H. Kim. Parallel methods and tools for predicting material properties. *Computing in Science & Engineering* **2**, 19 (2000).

[42] D. Sundholm. Universal method for computation of electrostatic potentials. *J. Chem. Phys.* **122**, 194107 (2005).

[43] I. Vasiliev, S. Ogut and J. R. Chelikowsky. First-principles density-functional calculations for optical spectra of clusters and nanocrystals. *Phys. Rev. B* **65**, 115416 (2002).

[44] A. Nicholls and B. Honig. A rapid finite-difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. *J. Comp. Chem.* **12**, 435 (1991).

[45] N. Baker, D. Sept, M. J. Holst and J. A. McCammon. The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. *IBM J. Res. Dev.* **45**, 427 (2001).

[46] M. Holst, J. McCammon and Z. Yu. Adaptive finite element modeling techniques for the Poisson-Boltzmann equation. *Commun. Comput. Phys.* **11**, 179 (2012).

[47] K. Singer. The Use of Gaussian (Exponential Quadratic) Wave Functions in Molecular Problems. I. General Formulae for the Evaluation of Integrals. *Proc. R. Soc. A* **258**, 412 (1960).

[48] L. E. McMurchie and E. R. Davidson. One- and two-electron integrals over Cartesian Gaussian functions. *J. Comp. Phys.* **26**, 218 (1978).

[49] S. Obara and A. Saika. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **84**, 3963 (1986).

[50] S. Obara and A. Saika. General recurrence formulas for molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **89**, 1540 (1988).

[51] M. Dupuis, J. Rys and H. F. King. Evaluation of molecular integrals over Gaussian basis functions. *J. Chem. Phys.* **65**, 111 (1976).

[52] M. Abramowitz and I. Stegun, eds. *Handbook of Mathematical Functions*, (Dover, New York, 1965).

[53] A. H. Boschitsch, M. O. Fenley and W. K. Olson. A Fast Adaptive Multipole Algorithm for Calculating Screened Coulomb (Yukawa) Interactions. *J. Comp. Phys.* **151**, 212 (1999).

[54] H. Yukawa. On the interaction of elementary particles. *Proc. Phys. Math. Soc. Japan* **17**, 48 (1935).

[55] J. Schwinger. On the bound states of a given potential. *Proc. Natl. Acad. Sci. USA* **47**, 122 (1961).

[56] M. H. Kalos. Monte Carlo Calculations of the Ground State of Three- and Four-Body Nuclei. *Phys. Rev.* **128**, 1791 (1962).

[57] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan and G. Beylkin. Multiresolution quantum chemistry: basic theory and initial applications. *J. Chem. Phys.* **121**, 11587 (2004).

[58] T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison and G. Beylkin. Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for Hartree–Fock and density functional theory. *J. Chem. Phys.* **121**, 2866 (2004).

[59] F. A. Bischoff and E. F. Valeev. Low-order tensor approximations for electronic wave functions: Hartree-Fock method with guaranteed precision. *J. Chem. Phys.* **134**, 104104 (2011).

[60] L. Frediani, E. Fossgaard, T. Flå and K. Ruud. Fully adaptive algorithms for multivariate integral equations using the non-standard form and multiwavelets with applications to the Poisson and bound-state Helmholtz kernels in three dimensions. *Mol. Phys.* **111**, 1143 (2013).

[61] D. Griffiths. *Introduction to Quantum Mechanics*, (Pearson Prentice Hall, New Jersey, 2005).

[62] T. Kato. On the eigenfunctions of many-particle systems in quantum mechanics. *Commun. Pur. Appl. Math.* **10**, 151 (1957).

[63] M. Morrell, R. Parr and M. Levy. Calculation of ionization potentials from density matrices and natural functions, and the long-range behavior of natural orbitals and electron density. *J. Chem. Phys.* **62**, 549 (1975).

[64] J. Katriel and E. Davidson. Asymptotic behavior of atomic and molecular wave functions. *Proc. Natl. Acad. Sci. USA* **77**, 4403 (1980).

[65] R. F. W. Bader. A quantum theory of molecular structure and its applications. *Chem. Rev.* **91**, 893 (1991).

[66] D. P. Carroll, H. J. Silverstone and R. M. Metzger. Piecewise polynomial configuration interaction natural orbital study of $1s^2$ helium. *J. Chem. Phys.* **71**, 4142 (1979).

[67] W. Klopper and W. Kutzelnigg. Gaussian basis sets and the nuclear cusp problem **135**, 339 (1986).

[68] W. Kutzelnigg and J. D. Morgan. Rates of convergence of the partial-wave expansions of atomic correlation energies. *J. Chem. Phys.* **96**, 4484 (1992).

[69] J. C. Slater. Atomic Shielding Constants. *Phys. Rev.* **36**, 57 (1930).

[70] S. F. Boys. Electronic Wave Functions. I. A General Method of Calculation for the Stationary States of Any Molecular System. *Proc. R. Soc. A* **200**, 542 (1950).

[71] J. Junquera, O. Paz, D. Sánchez-Portal and E. Artacho. Numerical atomic orbitals for linear-scaling calculations. *Phys. Rev. B* **64**, 235111 (2001).

[72] T. Ozaki and H. Kino. Numerical atomic basis orbitals from H to Kr. *Phys. Rev. B* **69**, 195113 (2004).

[73] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter and M. Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals **180**, 2175 (2009).

[74] C. C. J. Roothaan. New Developments in Molecular Orbital Theory. *Rev. Mod. Phys.* **23**, 69 (1951).

[75] G. G. Hall. The Molecular Orbital Theory of Chemical Valency. VIII. A Method of Calculating Ionization Potentials. *Proc. R. Soc. Lond. A* **205**, 541 (1951).

[76] P. Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Letters* **73**, 393 (1980).

[77] M. Head-Gordon and J. A. Pople. Optimization of wave function and geometry in the finite basis Hartree-Fock method. *J. Phys. Chem.* **92**, 3063 (1988).

[78] L. Thøgersen, J. Olsen, D. Yeager, P. Jørgensen, P. Sałek and T. Helgaker. The trust-region self-consistent field method: Towards a black-box optimization in Hartree–Fock and Kohn–Sham theories. *J. Chem. Phys.* **121**, 16 (2004).

[79] J. Almlöf, K. Fægri and K. Korsell. Principles for a direct SCF approach to LCAO-MO ab-initio calculations. *J. Comp. Chem.* **106**, 10891 (1982).

[80] M. Häser and R. Ahlrichs. Improvements on the direct SCF method. *J. Comp. Chem.* **10**, 104 (1989).

[81] J. Kussmann and C. Ochsenfeld. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **138**, 134114 (2013).

[82] R. W. Nunes and D. Vanderbilt. Generalization of the density-matrix method to a nonorthogonal basis. *Phys. Rev. B* **50**, 17611 (1994).

[83] A. M. N. Niklasson. Expansion algorithm for the density matrix. *Phys. Rev. B* **66**, 155115 (2002).

[84] W. J. Hehre, R. F. Stewart and J. A. Pople. Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals. *J. Chem. Phys.* **51**, 2657 (1969).

[85] M. Head-Gordon, J. A. Pople and M. J. Frisch. MP2 energy evaluation by direct methods. *Chem. Phys. Letters* **153**, 503 (1988).

[86] R. Lindh, U. Ryu and B. Liu. The reduced multiplication scheme of the Rys quadrature and new recurrence relations for auxiliary function based two-electron integral evaluation. *J. Chem. Phys.* **95**, 5889 (1991).

[87] P. M. W. Gill and J. A. Pople. The prism algorithm for two-electron integrals. *Int. J. Quant. Chem.* **40**, 753 (1991).

[88] J. Thom H. Dunning. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J. Chem. Phys.* **90**, 1007 (1989).

[89] P. Wind, T. Helgaker and W. Klopper. Efficient evaluation of one-center three-electron Gaussian integrals. *Theo. Chem. Acc.* **106**, 280 (2001).

[90] Nelder, J. A. and Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **7**, 308 (1965).

[91] E. Jones, T. Oliphant, P. Peterson et al. SciPy: Open source scientific tools for Python (2001–).

[92] M. W. Schmidt and K. Ruedenberg. Effective convergence to complete orbital bases and to the atomic Hartree–Fock limit through systematic sequences of Gaussian primitives. *J. Chem. Phys.* **71**, 3951 (1979).

[93] C. Froese Fischer. *The Hartree-Fock Method for Atoms: A Numerical Approach*, (John Wiley and Sons, New York , 1977).

[94] C. Froese Fischer. Multi-Configuration Hartree-Fock Program with Improved Stability. *Comput. Phys. Commun.* **4**, 107 (1972).

[95] E. A. McCullough Jr. Seminumerical SCF Calculations on Small Diatomic-Molecules. *Chem. Phys. Letters* **24**, 55 (1974).

[96] E. A. McCullough Jr. Partial-Wave Self-Consistent-Field Method for Diatomic-Molecules - Computational Formalism and Results for Small Molecules. *J. Chem. Phys.* **62**, 3991 (1975).

[97] L. Laaksonen, P. Pyykkö and D. Sundholm. Two-dimensional fully numerical solutions of molecular Schrödinger equations. I. One-electron molecules. *Int. J. Quant. Chem.* **23**, 309 (1983).

[98] L. Laaksonen, P. Pyykkö and D. Sundholm. Two-Dimensional fully numerical solutions of molecular Schrödinger equations. II. Solution of the Poisson equation and results for singlet states of H_2 and HeH^+. *Int. J. Quant. Chem.* **23**, 319 (1983).

[99] L. Laaksonen, P. Pyykkö and D. Sundholm. Two-dimensional fully numerical-solutions of molecular Hartree-Fock equations - LiH and BH. *Chem. Phys. Letters* **96**, 1 (1983).

[100] D. Sundholm and J. Olsen. Large MCHF Calculations on the Hyperfine Structure of $B(^2P)$ and the Nuclear Quadrupole Moments of ^{10}B and ^{11}B. *J. Chem. Phys.* **94**, 5051 (1991).

[101] J. Kobus, L. Laaksonen and D. Sundholm. A numerical Hartree-Fock program for diatomic molecules. *Comput. Phys. Commun.* **98**, 346 (1996).

[102] J. R. Chelikowsky, N. Troullier and Y. Saad. Finite-difference-pseudopotential method: Electronic structure calculations without a basis. *Phys. Rev. Letters* **72**, 1240 (1994).

[103] T. Torsti, T. Eirola, J. Enkovaara, T. Hakala, P. Havu, V. Havu, T. Höynälänmaa, J. Ignatius, M. Lyly, I. Makkonen, T. T. Rantala, J. Ruokolainen, K. Ruotsalainen, E. Räsänen, H. Saarikoski and M. J. Puska. Three real-space discretization techniques in electronic structure calculations. *Phys. Status Solidi (b)* **243**, 1016 (2006).

[104] A. Castro, H. Appel, M. Oliveira, C. A. Rozzi, X. Andrade, F. Lorenzen, M. A. L. Marques, E. K. U. Gross and A. Rubio. octopus: a tool for the application of time-dependent density functional theory. *Phys. Status Solidi (b)* **243**, 2465 (2006).

[105] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Mø ller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nø rskov, M. Puska, T. T. Rantala, J. Schiø tz, K. S. Thygesen and K. W. Jacobsen. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *J. Phys.: Condens. Matter* **22**, 253202 (2010).

[106] A. D. Becke. Basis-set-free density-functional quantum chemistry. *Int. J. Quant. Chem.* **36**, 599 (1989).

[107] L. Lehtovaara, V. Havu and M. Puska. All-electron time-dependent density functional theory with finite elements: time-propagation approach. *J. Chem. Phys.* **135**, 154104 (2011).

[108] P. Čársky. Efficient evaluation of Coulomb integrals in a mixed Gaussian and plane-wave basis. *Int. J. Quant. Chem.* **107**, 56 (2007).

[109] M. A. Watson, Y. Kurashige, T. Nakajima and K. Hirao. Linear-scaling multipole-accelerated Gaussian and finite-element Coulomb method. *J. Chem. Phys.* **128**, 054105 (2008).

[110] D. Sundholm, P. Pyykkö and L. Laaksonen. Two-dimensional, fully numerical molecular calculations. VIII. Electric field gradients of diatomic hydrides LiH-ClH at the HFS level. *Mol. Phys.* **55**, 627 (1985).

[111] E. Van Lenthe and E. J. Baerends. Optimized Slater-type basis sets for the elements 1–118. *J. Comp. Chem.* **24**, 1142 (2003).

[112] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. *SIAM Review* **51**, 455 (2009).

[113] J. Jusélius and D. Sundholm. Parallel implementation of a direct method for calculating electrostatic potentials. *J. Chem. Phys.* **126**, 094101 (2007).

[114] I. S. Ufimtsev and T. J. Martinez. Quantum chemistry on graphical processing units. 1. strategies for two-electron integral evaluation **4**, 222 (2008).

Appendix

Paper I

I

I

The direct approach to gravitation and electrostatics method for periodic systems

S. A. Losilla,[1] D. Sundholm,[1,a)] and J. Jusélius[2]
[1]*Department of Chemistry, University of Helsinki, FIN-00014 Helsinki, Finland*
[2]*Department of Chemistry, University of Tromsø, N-9037 Tromsø, Norway*

(Received 5 November 2009; accepted 18 December 2009; published online 11 January 2010)

The direct approach to gravitation and electrostatics (DAGE) algorithm is an accurate, efficient, and flexible method for calculating electrostatic potentials. In this paper, we show that the algorithm can be easily extended to consider systems with many different kinds of periodicities, such as crystal lattices, surfaces, or wires. The accuracy and performance are nearly the same for periodic and aperiodic systems. The electrostatic potential for semiperiodic systems, namely defects in crystal lattices, can be obtained by combining periodic and aperiodic calculations. The method has been applied to an ionic model system mimicking NaCl, and to a corresponding covalent model system. © *2010 American Institute of Physics*. [doi:10.1063/1.3291027]

I. INTRODUCTION

The charge density of periodic systems can be expressed as an infinite sum of the density of repeated unit cells. This view provides a possibility to extend direct integration approaches to studies of surfaces, two-dimensional (2D) slabs, and one-dimensional (1D) wires. The periodic bulk material has an infinite number of unit cells, each associated with a vector **k**, whose integer components in three dimensions k_x, k_y, and k_z formally run from $-\infty$ to ∞. For slabs, the summation of the unit cells in the z direction consists of only one term, whereas the two remaining dimensions are formally infinite. Analogously, a wire is infinite in only one dimension and finite in the two others. A completely aperiodic object or a general molecule is considered in only one unit cell.

The periodicity can also be truncated in one direction of a periodic dimension giving raise to semiperiodic objects such as surfaces, edges, or corners of the aforementioned systems. For surfaces, one dimension (e.g., z) is truncated with k_z taking integer values in the interval $[0,\infty[$. For more realistic cases, the density for such systems will not be truly periodic, as the density at the edges is different from the density in the bulk. However, the system can still be considered as a combination of two or more systems, each one of them with their own periodicity. For instance, a surface can be considered as one or more 2D infinite slabs forming the upper layers, on top of a truncated three-dimensional (3D) crystal lattice. This view also allows for introducing aperiodic perturbations in a periodic structure, which can be found in systems such as defects in the solid state or clusters embedded in a periodic matrix.

Calculation of electrostatic interactions for periodic systems is of great importance for many applications. A variety of well-established methods are available for this purpose. For 3D periodic systems, the Ewald summation method is the classical approach for calculating electrostatic interactions between point charges in periodic systems.[1] For periodic systems with smooth electron densities the fast Fourier transformation method is a powerful approach, because the Laplacian is diagonal in the momentum representation, and the Fourier transform to the reciprocal space scales well with the size of the system. For large finite aperiodic systems, an efficient method to solve electrostatic interaction problems is the fast multipole method.[2–5]

For aperiodic perturbations in periodic systems, the solution of the electrostatic problem is more complicated because the method should be able to consider the periodic structure of the bulk material and the aperiodicty due to defects or sparsely embedded molecules and clusters. Two main approaches have been employed to tackle such systems. The straightforward approach is to assume that the system is periodic even though it contains defects interrupting the periodicity. The main disadvantage with such an approach is that the impurities are too dense, giving rise to an overestimation of the interactions between them. The spurious defect-defect interactions can be reduced by using larger supercells rendering the calculations computationally more expensive.[6] Alternatively, embedding schemes can be employed. Then, the quantum mechanical calculation is performed without periodic boundary conditions and the interactions of the surrounding are simulated by using point charges or more realistic charge densities.[7,8] The embedded scheme at the point-charge level has very recently been extended to treat molecules embedded in a periodic environment.[9]

Here, we present an extension of the DAGE method[10,11] to periodic systems. We show that the original algorithm can be generalized in a very straightforward manner, allowing calculations of electrostatic potentials for many types of periodic systems. The method can be extended to semiperiodic systems. The main features, such as sub-ppm accuracy, nearly linear computational scaling and the possibility to achieve linear scaling,[5] and good parallel scaling are retained.

By dividing the charge density into a periodic and a

a)Electronic mail: sundholm@chem.helsinki.fi.

defect density, the present approach can be used to calculate the electrostatic potential for a single defect or for a molecule embedded by a periodic matrix. Natan *et al.*[12] and Genovese and co-workers[13–15] recently developed similar methods to calculate the electrostatic potential for surfaces and aperiodic systems. They use local basis functions to describe the functions in the aperiodic direction, whereas in the two periodic dimensions the density is expanded in plane waves.[14] They employ wavelets as local basis functions as also Fann *et al.*[16] Harrison and co-workers[17–19] do in their MADNESS program for numerical electronic structure calculations.

This article is structured as follows. The basic theory and a presentation of the general algorithm, including the treatment of periodic systems, are given in Sec. II. Section III discusses some important numerical aspects on the treatment of the r_{12}^{-1} operator. Calculations of the electrostatic potential for two model densities are presented in Sec. IV. The methods and main results are summarized in Sec. V.

II. THEORY

The electrostatic potential corresponding to a given charge density can be obtained by calculating the Coulomb integral

$$V(\mathbf{r}_1) = \int_\Omega \rho(\mathbf{r}_2)\frac{1}{r_{12}}d^3r_2 \tag{1}$$

over the whole 3D domain Ω. For periodic systems, the charge densities of all unit cells are identical to the density of a reference unit cell, $\rho_0(\mathbf{r})$

$$\rho_{\mathbf{k}}(\mathbf{r}) = \rho_0(\mathbf{r} - \mathbf{k}\cdot\mathbf{R}). \tag{2}$$

In Eq. (2), the vector components of $\mathbf{R}$, R_x, R_y, and R_z are the dimensions of the unit cell in the Cartesian directions. Thus, $\mathbf{k}\cdot\mathbf{R}$ is the offset between the reference and the $\mathbf{k}$th unit cell. The charge density of a periodic system expressed as a sum of contributions from locally defined unit cells becomes

$$\rho(\mathbf{r}) = \sum_{\mathbf{k}} \rho_0(\mathbf{r} - \mathbf{k}\cdot\mathbf{R}). \tag{3}$$

This implies that the electrostatic potential in a given unit cell can also be obtained as a sum of the contributions from all unit cells. The expression for the electrostatic potential in the reference unit cell can then be written as

$$V_0(\mathbf{r}_1) = \sum_{\mathbf{k}} \int_\Omega \rho_0(\mathbf{r}_2)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2 - \mathbf{k}\cdot\mathbf{R}|}d^3r_2. \tag{4}$$

By rewriting the Coulomb operator as an integral expression using the well-known identity

$$\frac{1}{r} = \frac{2}{\sqrt{\pi}}\int_0^\infty e^{-t^2r^2}dt \tag{5}$$

discretizing the auxiliary t integral, and calculating it numerically using, e.g., Gaussian quadrature, the Coulomb expression for the electrostatic potential in the unit cell becomes

$$V_0(x_1,y_1,z_1) = \frac{2}{\sqrt{\pi}}\sum_p \omega_p \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{k_xk_yk_z} e^{-t_p^2(x_2 + k_xR_x - x_1)^2} \times e^{-t_p^2(y_2 + k_yR_y - y_1)^2}e^{-t_p^2(z_2 + k_zR_z - z_1)^2} \times \rho_0(x_2,y_2,z_2)dx_2dy_2dz_2, \tag{6}$$

where the integration points t_p and the corresponding integration weights ω_p have been introduced. The density and the potential in the reference unit cell can be discretized by expanding them using tensorial finite-element (FE) functions. In this work, we use Lagrange interpolation polynomials (LIPs) as numerical basis functions. Other representations that have been used in this context are wavelets and Chebyshev functions.[5,15] The main advantage of tensorial numerical basis functions is the separability; the integrals of the multidimensional case can be expressed as outer products of the 1D ones rendering efficient computer implementations easier.[20]

In the 1D case, the LIP representation of a function $\phi(x)$ is given by

$$\phi(x) = \sum_i f_i\chi_i(x), \tag{7}$$

where $\chi_i(x)$ are the 1D interpolation functions

$$\chi_i(x) = \prod_{l\neq i}\frac{(x - x_l)}{(x_i - x_l)} \tag{8}$$

and f_i are the corresponding expansion coefficients. The x_l points are often denoted grid or mesh points making a natural connection between the present FE method and finite-difference approaches.[21,22] In Eq. (8), one can see that $\chi_i(x)$ have the property of being one in one of the x_l points and zero in all the others. This implies that when a given function is expanded in LIPs, its expansion coefficients, f_i, are equal to the value of the function in the x_l points. One can take advantage of this property by approximating functionals by projecting the function onto the LIP basis. For example, the charge density contribution from an orbital can be calculated as the square of the LIP expansion coefficients of the orbital. This property is preserved for higher dimensional spaces when tensorial basis functions are used.

$$\chi_{ijk}(x,y,z) = \chi_i(x)\chi_j(y)\chi_k(z). \tag{9}$$

In Eq. (9), the tensorial basis functions $\chi_{ijk}(x,y,z)$ are written as an outer product of the 1D LIP basis functions. The unit cell is divided into cubical elements. The density and the potential are expanded in the tensorial LIP functions in each element, whereas outside a given element the basis functions vanish. The continuity conditions are fulfilled by having the same expansion coefficients for the nonvanishing LIPs at the element borders.

As in the aperiodic case, the LIP expansion coefficients of the potential ($v_{\alpha\beta\gamma}$) in the reference unit cell can be obtained from the corresponding expansion coefficients of the density ($d_{\alpha'\beta'\gamma'}$) as

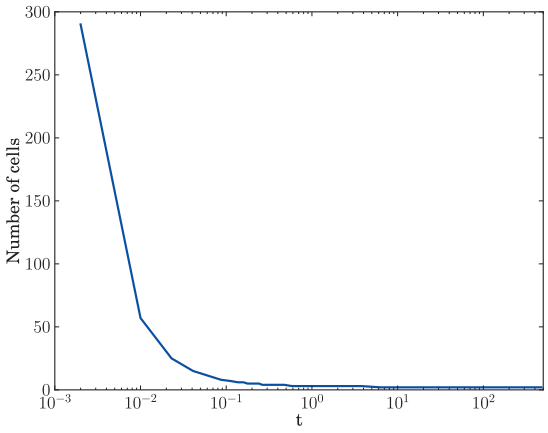


FIG. 1. The maximum number of cells required for the summation in Eq. (11) to converge to a relative accuracy of 10^{-12}. The volume of the cubic unit cell is 10.6^3 bohr^3. Sixth order LIP are used as element functions.

$$v_{\alpha\beta\gamma} = \frac{2}{\sqrt{\pi}} \sum_{p}^{p_{\max}} \omega_p \sum_{\gamma'}^{N_z} F^{z,p}_{\gamma\gamma'} \sum_{\beta'}^{N_y} F^{y,p}_{\beta\beta'} \sum_{\alpha'}^{N_x} F^{x,p}_{\alpha\alpha'} d_{\alpha'\beta'\gamma'}, \qquad (10)$$

where the $\mathbf{F^p}$ matrices are given by

$$F^{x,p}_{\alpha\alpha'} = \sum_{k_x} \int_{-\infty}^{\infty} e^{-t_p^2 (x_\alpha - x_2 - k_x R_x)^2} \chi_{\alpha'}(x_2) dx_2. \qquad (11)$$

The formal integration from $-\infty$ to ∞ in Eq. (11) is limited to the element where $\chi_{\alpha'}(x_2)$ is defined. The x_α denote coordinates of the grid points in one dimension where the potential is to be calculated.[5,10,11] Notice that the grid points for the potential x_α and the grid points for the density x'_α do not need to be the same. Analogous expressions can be obtained for the $\mathbf{F^{y,p}}$ and $\mathbf{F^{z,p}}$ matrices.

The only difference between the periodic and aperiodic cases is the formally infinite summation of the contributions to the integrals of the $\mathbf{F}^{x,p}$, $\mathbf{F}^{y,p}$, and $\mathbf{F}^{z,p}$ matrices. This summation converges very fast for t-values larger than 1. Figure 1 shows the maximum number of terms needed to obtain a relative accuracy of 10^{-12} respect to the highest term in the sum. Very small t-values require that many cells are considered, but as seen in the present applications, they can be omitted as they hardly contribute to the total potential.

By performing the sum over unit cells in only one or two dimensions, wires or planes can be studied. The approach can also be generalized to other periodic objects, such as edges or corners, as long as the whole set of $\mathbf{k}$ vectors can be constructed as an outer product of 1D translation vectors, i.e., $\{\mathbf{k}\}=\{k_x\}\otimes\{k_y\}\otimes\{k_z\}$. Note that the charge density in an unit cell at an edge is usually not exactly the same as the density in an unit cell in the bulk. The total density can hence not be decomposed into an infinite sum of shifted reference cells. However, the final potential can be obtained as a sum of two or more contributions from different densities, each one with its own periodicity.

Once the $\mathbf{F^p}$ matrices are constructed, the algorithm is identical for the periodic and aperiodic cases. The evaluation of the potential using Eq. (10) consists of three sets of coupled matrix multiplications for each t_p value and for one of the spatial indices (α, β, or γ). The final expansion coefficients of the potential are obtained by adding each t_p contribution multiplied by the appropriate integration weight. Computational aspects of the algorithm have been discussed in our previous works.[10,11]

III. INTEGRATION IN THE *T*-SPACE

The accuracy of the calculated potentials is determined by two factors: the tightness of the numerical grid in the three Cartesian dimensions, and the quality of the integration in the t-space. The grid can be systematically improved by increasing the number of points until the results converge to the desired accuracy. The numerical integration in the t-space requires a more careful sampling. Although adding more quadrature points would eventually yield accurate results as well, such an approach is inefficient, and it is difficult to assess the quality of the obtained results. A better understanding of the integrand is crucial in order to improve the integration scheme.

The expression for the potential can be rewritten as

$$V(\mathbf{r_1}) = \frac{2}{\sqrt{\pi}} \int_0^\infty \left[\int_\Omega \rho(\mathbf{r_2}) e^{-t^2(\mathbf{r_1}-\mathbf{r_2})^2} d^3 r_2 \right] dt$$
$$= \frac{2}{\sqrt{\pi}} \int_0^\infty W(\mathbf{r_1}, t) dt. \qquad (12)$$

If $\rho(\mathbf{r_2})$ is continuous, $W(\mathbf{r_1},t)$ is continuous in t as well as in $\mathbf{r_1}$. When $t=0$, the Gaussian operator becomes the identity, and $W(\mathbf{r_1},0)$ is the total charge Q

$$W(\mathbf{r_1},0) = \int_\Omega \rho(\mathbf{r_2}) d^3 r_2 = Q. \qquad (13)$$

For neutral systems, $W(\mathbf{r_1},0)=0$. For small t-values, $W(\mathbf{r_1},t)$ is expected to be easily approximated with polynomials, which are accurately integrated using Gaussian quadrature.

For large t-values, a powerful integration scheme can be obtained by expanding the density $\rho(\mathbf{r_2})$ in a Taylor series around the coordinate $\mathbf{r_1}$

$$\rho(\mathbf{r_2}) = \rho(\mathbf{r_1}) + \sum_i \left(\frac{\partial \rho(\mathbf{r_2})}{\partial x_{2,i}} \right)_{\mathbf{r_1}} x_{12,i}$$
$$+ \frac{1}{2} \sum_{ij} \left(\frac{\partial}{\partial x_{2,i}} \frac{\partial \rho(\mathbf{r_2})}{\partial x_{2,j}} \right)_{\mathbf{r_1}} x_{12,i} x_{12,j} + \mathcal{O}(\mathbf{r}_1^3) \qquad (14)$$

which is correct up to second order. The corresponding series expansion for $W(\mathbf{r_1},t)$ then becomes

$$W(\mathbf{r_1},t) = \rho(\mathbf{r_1}) \frac{\pi^{(3/2)}}{t^3} + \frac{1}{4} (\nabla^2 \rho(\mathbf{r_2}))_{\mathbf{r_1}} \frac{\pi^{(3/2)}}{t^5} + \mathcal{O}(t^{-7}). \qquad (15)$$

For sufficiently large t-values the Gaussian operator is sharp, and can be considered as a good approximation to the Dirac δ function centered at $\mathbf{r_1}$. This allows separating the integral in Eq. (12) into two parts

I

TABLE I. Errors in the numerical integration of $\int_1^a x^{-3}dx$ (in ppm) using 12-point linear and logarithmic Gaussian quadratures.

a	Linear quadrature	Logarithmic quadrature
2	-1.5×10^{-10}	-1.1×10^{-10}
5	-3.0×10^{-2}	-2.3×10^{-10}
10	-3.2×10^{1}	-4.5×10^{-10}
10^2	-2.7×10^{5}	-7.8×10^{-10}
10^4	-1.0×10^{6}	-8.4×10^{-5}
10^6	-1.0×10^{6}	-5.6×10^{-2}

$$V(\mathbf{r_1}) = \frac{2}{\sqrt{\pi}}\int_0^{t_f} W(\mathbf{r_1},t)dt + \frac{\pi}{t_f^2}\rho(\mathbf{r_1}) + \mathcal{O}(t_f^{-4}). \tag{16}$$

The first integral still has to be solved numerically. The error in the potential at each point of space due to omission of the second order term in Eq. (15) is

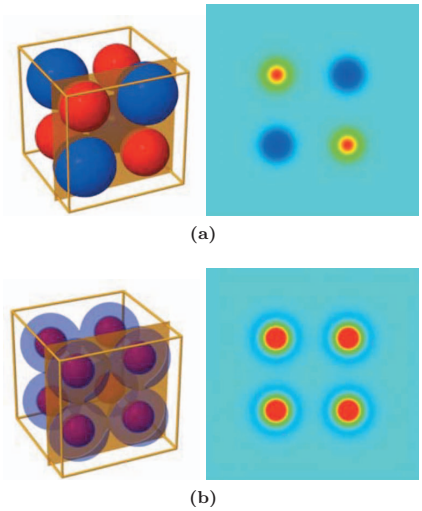
$$\varepsilon_V^{(2)}(\mathbf{r_1}) = \frac{\pi}{8t_f^4}(\nabla^2\rho(\mathbf{r_2}))_{\mathbf{r_1}}. \tag{17}$$

Even when this approximation does not hold for the whole domain, $W(\mathbf{r_1},t)$ will nevertheless be a slowly decaying function for large t-values. This kind of functions is difficult to interpolate accurately using polynomials. Gaussian quadrature is thus expected to perform poorly for integrating $W(\mathbf{r_1},t)$ in that region. An alternative quadrature can be obtained by integrating in logarithmic coordinates. The transformed integral can then be calculated numerically

$$\begin{aligned}\int_{t_l}^{t_f} W(\mathbf{r_1},t)dt &= \int_{\log(t_l)}^{\log(t_f)} W(\mathbf{r_1},e^s)e^s ds \\ &\approx \sum_p \omega_p W(\mathbf{r_1},e^{s_p})e^{s_p} \\ &= \sum_p \beta_p W(\mathbf{r_1},u_p).\end{aligned} \tag{18}$$

The accuracy of both quadratures is assessed by computing the errors for the numerical calculation of $\int_1^a x^{-3}dx$ for different values of a. The results are shown in Table I. While standard Gaussian quadrature would require a huge number of points to accurately integrate $W(\mathbf{r_1},t)$ over several orders of magnitude, the Gaussian quadrature in the logarithmic scale needs only a few integration points to yield very small numerical errors.

The integration domain $[0,\infty[$ is divided in three regions. The linear region $[0,t_l]$ is integrated using Gaussian quadrature, the logarithmic region $[t_l,t_f]$ using Gaussian quadrature in logarithmic coordinates, and the tail region $[t_f,\infty[$ is integrated using the second term of Eq. (16). An accurate integration scheme requires a careful choice of t_l and t_f. A good value for t_f can be estimated using Eq. (17). The optimal t_l is the smallest t-value for which $W(\mathbf{r_1},t_f)$ is proportional to t_f^{-3} for the whole $\mathbf{r_1}$ space. A simple way to find this parameter is to examine the average behavior of $W(\mathbf{r_1},t)$. This can be achieved by expanding the self-interaction Coulomb energy E in t and integrating over $\mathbf{r_1}$ for each t-value as



(a)

(b)

FIG. 2. Charge densities of the (a) ionic and the (b) covalent systems. The isosurfaces are plotted at 0.02 (red) and −0.02 (blue). Planar cuts of the charge densities are also shown.

$$\begin{aligned}E &= \int_\Omega V(\mathbf{r_1})\rho(\mathbf{r_1})d^3r_1 \\ &= \frac{2}{\sqrt{\pi}}\int_0^\infty \left[\int_\Omega W(\mathbf{r_1},t)\rho(\mathbf{r_1})d^3r_1\right]dt \\ &= \frac{2}{\sqrt{\pi}}\int_0^\infty F(t)dt.\end{aligned} \tag{19}$$

t_l can then be easily estimated by plotting $F(t)$. Once t_l and t_f are set, the number of quadrature points can be increased until a probe quantity (e.g., the self-interaction energy) is converged to the desired accuracy, or preferably, until the precision does not improve by increasing the number of integration points, i.e., the remaining error is only due to the quality of the Cartesian grid.

IV. CASE STUDIES

Two model systems have been considered for testing the algorithm: an "ionic" and a "covalent" one. The unit cell of the ionic model system, shown in Fig. 2(a), consists of eight Gaussian charge distributions, of which four have a charge of $+e$ and four have the opposite charge $-e$. The centers of the Gaussians form a cube with side lengths of 5.3 a.u., similar to the NaCl rock salt structure. The exponents are 1.2 for the positive Gaussians and 0.6 for the negative ones. They have been chosen to yield a charge of $0.99e$ when the densities are integrated up to the ionic radii of Na^+ and Cl^-. The total net charge is zero. The covalent system, shown in Fig. 2(b), has the same structure, but in this case every lattice site has two Gaussians with opposite sign. The resulting charge density is composed of positively charged cores surrounded by negatively charged shells, also with no net charge.

The model systems have been chosen for a number of reasons. As the densities are composed by Gaussian func-

tions, analytical values are very easy to compute. The charge densities are smooth, and should be handled accurately using finite elements. For periodic systems, a zero net charge is required in every cell. Otherwise, the total charge of the system would become infinite. From the computational point of view, the most important difference between the two systems is that the ionic one has local dipoles, while the covalent one does not. The ionic system cannot be exactly separated into a product of 1D functions, while the covalent can. This provides additional evidence for the validity of the tensorial approach.

For these model systems, the Gaussian charge distributions extend outside the boundaries of the unit cell. The calculation domain has to be increased in the aperiodic dimension to contain the entire charge. A box that is 1.5 times larger than the unit cell in each dimension contains the whole charge, up to ppm accuracy. For the periodic cases, the charge density, defined by Eq. (3), is completely contained inside the unit cell. Therefore, for aperiodic dimensions, the calculation domain is 15.9 bohr long, whereas for periodic dimensions it has the same size as the unit cell (10.6 bohr). In order to keep the same step length between the grid points in the two cases and to achieve a comparable accuracy, the number of finite elements per dimension has to be accordingly increased.

The numerical grids are denoted here by the step length in bohr. For instance, the 0.146 grid is a grid with 109 grid points in aperiodic dimensions and 73 grid points in periodic dimensions. This corresponds to 18 or 12 finite elements for the sixth order 1D LIP basis. For the rest of the work, we focus mainly on optimizing the t-integration for the 0.146 grid. Calculations employing this grid size are not computationally expensive, taking about 50 s on an AMD64 Dual Core 4600+ workstation for the completely aperiodic case.

Figures 3(a) and 3(b) show the $F(t)$ curves for the ionic and the covalent systems, respectively, calculated using free boundary conditions (P0D), and periodic boundary conditions along one (P1D), two (P2D) or three (P3D) Cartesian axes with the 0.146 grid. The $F(t)$ curves behave as discussed in Sec. III. For $t \geq 2$, the curves decay toward zero such as t^{-3}, implying that t_l can be set to 2. For small values of t, the curves decay to $Q=0$, as the total densities do not contain any net charge. For the P3D cases, $F(t)$ unexpectedly increases at small t-values. Calculations using a denser grid showed that this is an artifact due to the use of a too small grid. For the P3D cases, the problematic small t-values contribute very little to the potential, implying that they can be disregarded for the P3D periodic systems. The first integration point t_i should be chosen where $F(t_i)$ is small enough and insensitive to grid refinements. The appropriate choices are 0.1 for the ionic system and 0.12 for the covalent one. The final integration point can be estimated from Eq. (17). In the current applications, $t_f=500$ introduces errors which are smaller than the overall accuracy threshold.

Two probe quantities are used to assess the accuracy of the t-integration, namely the self-interaction energy, and the potential along the line passing through the center of two of the Gaussian functions. The calculation of analytical reference values requires the explicit contribution from every unit cell image. The analytical self-interaction energy can be calculated for every case, but the analytical potentials for the cases with two and three periodic dimensions are too demanding to be computed.

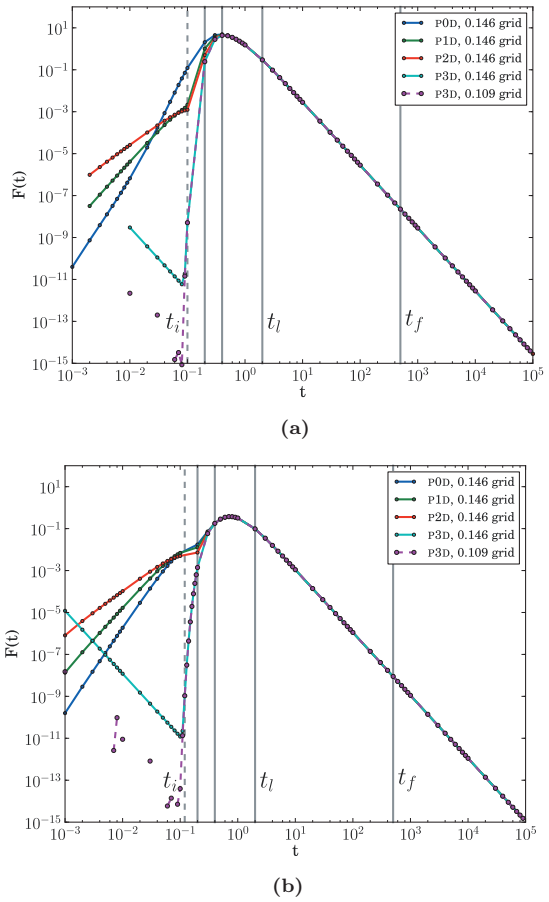


FIG. 3. $F(t)$ curves for (a) the ionic and (b) the covalent systems. The black vertical lines show the Gaussian quadrature intervals chosen for the final calculation. The dashed line is the lower integration limit t_i for the P3D case; for the rest of the cases $t_i=0$ was chosen. The interval $[t_i,t_l]$ is subdivided into three 12-point quadrature intervals.

We perform the integration by dividing the linear and logarithmic regions in sub-intervals, which are integrated using a 12-point quadrature. The accuracy limit for the t-integration is reached for all the considered cases using the 0.146 grid with one interval for the logarithmic region and three intervals for the linear region, namely $[t_i,0.2]$, $[0.2,0.4]$, $[0.4,2]$. This is a total of 48 quadrature points, which is 60% the amount used in our previous work.[11] It should be noted that for the P0D cases one less interval, that is 36 points, can be used for the linear region without affecting the accuracy. The results do not change significantly when the intervals are slightly altered. The accuracy of the potentials obtained with this integration scheme is shown in Fig. 4. Although the numerical errors at the center of the steeper Gaussian functions seem large, the maximum error is in all cases below 10 ppm as compared to the average accu-

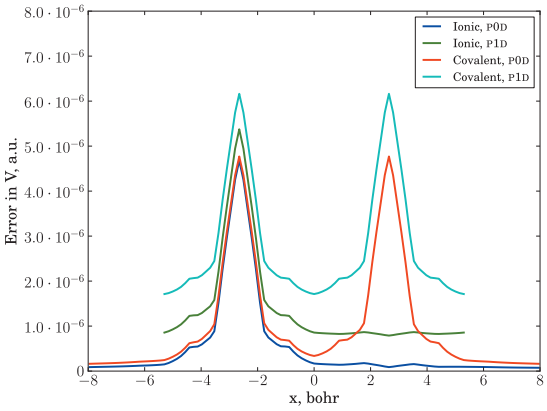


FIG. 4. Absolute error in the potential along a line parallel to the x axis and passing through the center of the Gaussian charge distributions obtained for the P0D and P1D cases using the 0.146 grid.

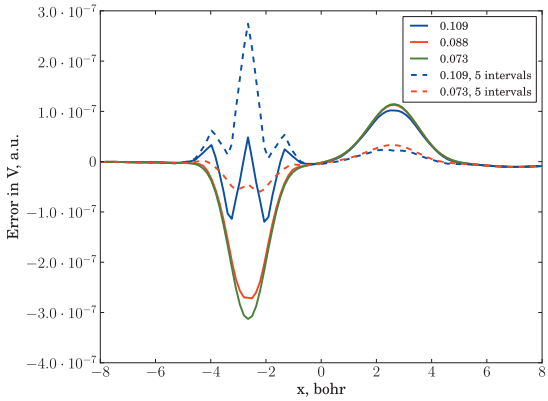


FIG. 5. Absolute error in the potential along a line parallel to the x axis and passing through the center of the Gaussian charge distributions for the P0D ionic system, calculated using different grids.

racy of around 1 ppm. For the periodic cases, the error is of the same magnitude as for the aperiodic ones, even though the baseline in the graph is slightly higher.

Table II shows the errors in the self-interaction energy calculated with the integration scheme described above. The errors decrease with increasing the number of elements per side of the integration domain. Sub-ppm accuracy is obtained with the 0.109 grid (145 and 109 grid points for the aperiodic and periodic cases, respectively). Increasing the number of elements further does not improve the accuracy, implying that the dominant remaining error is due to the integration in t space. Figure 5 shows that the tiny error in the self-interaction energy obtained with the 0.109 grid is due to error cancellations, as a larger average error is obtained when using a more accurate t-integration scheme.

As mentioned in Sec. II, the method can be extended for treating semiperiodic systems. The most simple cases are considered here, in which only the periodicity in one dimension is truncated, namely wire tips (S1P0D), slab edges (S1P1D), and crystal surfaces (S1P2D). As the density of each Gaussian extends only to the nearest neighboring unit cells, the total density of the model systems can be separated into two contributions, one from the tip, edge or surface, and one from the bulk. Cases with truncated periodicity in two or three dimensions, corresponding to corners, can be treated similarly, but the density needs to be decomposed into four or eight contributions. In realistic systems, the density will vary with the depth into the bulk in a more complicated way. This can be overcome by using an extended set of input charge densities, and by using larger supercells. Such calculations are outside the scope of the present work. Figure 6 shows the potentials of systems where the periodicity is truncated in zero or one dimensions.

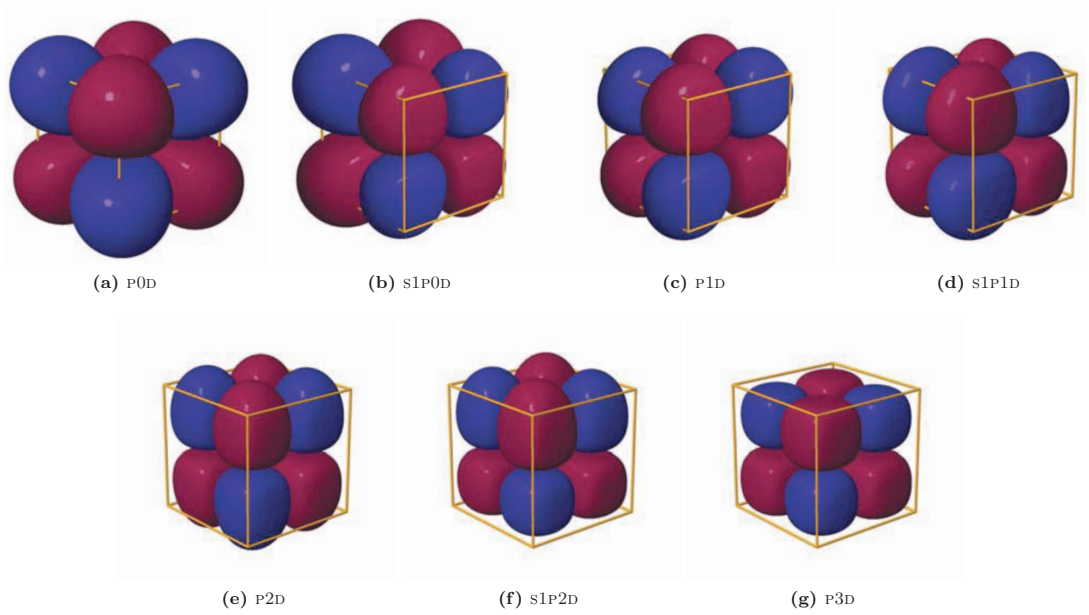
V. SUMMARY

The DAGE algorithm for efficient calculation of electrostatic potentials from given charge densities has been generalized to deal with periodic systems. The only difference compared to the aperiodic case appears in the construction of the $\mathbf{F}^{\mathbf{p}}$ matrices used for the linear transformations. For periodic systems, the matrices are obtained as a sum of contributions from spatially shifted images of the reference unit cell. For large t-values, the sum converges very fast, only a few neighboring cells have to be considered. For small t-values, also remote unit cells contribute to the $\mathbf{F}^{\mathbf{p}}$ matrices, increasing the computational costs for a given grid size. However, the contribution from very small t-values can be neglected in the P3D periodic calculations.

Once the $\mathbf{F}^{\mathbf{p}}$ matrices are constructed, they can be reused and employed in calculations of electrostatic potentials originating from different densities because the $\mathbf{F}^{\mathbf{p}}$ matrices are independent of the charge density. The computational costs, scaling behavior, and accuracy are the same for the linear

TABLE II. Errors in the self-interaction energy (in ppm) for the ionic and covalent systems calculated as a function of the grid step in bohr, obtained using different periodicities. The number of elements in the periodic/aperiodic dimensions is also given.

		Ionic system				Covalent system			
Step	Elements	P0D	P1D	P2D	P3D	P0D	P1D	P2D	P3D
0.427	4/6	1.38×10^{4}	1.45×10^{4}	2.20×10^{4}	1.50×10^{4}	6.67×10^{4}	7.04×10^{4}	2.58×10^{5}	6.36×10^{4}
0.217	8/12	-1.19×10^{2}	-1.22×10^{2}	-1.23×10^{2}	-1.25×10^{2}	-5.03×10^{2}	-5.03×10^{2}	-4.77×10^{2}	-4.68×10^{2}
0.146	12/18	-1.68×10^{0}	-1.77×10^{0}	-1.82×10^{0}	-1.85×10^{0}	5.27×10^{-2}	4.98×10^{-2}	4.68×10^{-2}	4.35×10^{-2}
0.109	16/24	1.26×10^{-1}	1.33×10^{-1}	1.44×10^{-1}	1.58×10^{-1}	-1.45×10^{1}	-1.45×10^{1}	-1.45×10^{1}	-1.44×10^{1}
0.088	20/30	1.59×10^{-1}	1.69×10^{-1}	1.80×10^{-1}	1.95×10^{-1}	3.97×10^{-1}	3.94×10^{-1}	3.92×10^{-1}	3.88×10^{-1}
0.073	24/36	1.63×10^{-1}	1.73×10^{-1}	1.85×10^{-1}	2.00×10^{-1}	4.38×10^{-1}	4.36×10^{-1}	4.33×10^{-1}	4.30×10^{-1}



(a) P0D (b) S1P0D (c) P1D (d) S1P1D

(e) P2D (f) S1P2D (g) P3D

FIG. 6. Potentials for the ionic system with different boundary conditions: (a) an isolated unit cell, (b) a tip unit cell of a wire, (c) a bulk unit cell of a wire, (d) an edge unit cell of a slab, (e) a bulk unit cell of a slab, (f) a surface unit cell of a crystal lattice, and (g) a bulk unit cell of a crystal lattice. The isosurfaces are plotted at 0.04 (red) and −0.04 (blue) a.u. The boundaries of the primitive unit cell are shown in yellow.

transformations in both the aperiodic and periodic cases. The calculation of the electrostatic potential for periodic systems might even be computationally cheaper because the charge density extends over a larger volume in the aperiodic case. For a given accuracy, the grid needed in the periodic calculations is therefore smaller than the one required by the aperiodic counterpart. The sources of error, especially the numerical treatment of the r_{12}^{-1} operator, were analyzed in detail. A cost-efficient integration scheme, applicable to general systems, has been developed.

ACKNOWLEDGMENTS

This research has been supported by the Academy of Finland through its Centers of Excellence Programme 2006–2011 and from the Nordic Centre of Excellence in Computational Chemistry (NCoECC) (NordForsk 070253). CSC, the Finnish IT Center for Science, is thanked for computer time. S.L. acknowledges CSC, AMD Inc., Cray Inc., and the Magnus Ehrnrooth's Foundation for financial support.

[1] P. Ewald, Ann. Phys. **369**, 253 (1921).
[2] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[3] L. Greengard, Science **265**, 909 (1994).
[4] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).
[5] M. A. Watson and K. Hirao, J. Chem. Phys. **129**, 184107 (2008).
[6] R. Orlando, R. Dovesi, C. Roetti, and V. R. Saunders, Chem. Phys. Lett. **228**, 225 (1994).
[7] T. A. Wesolowski and A. Warshel, J. Phys. Chem. **97**, 8050 (1993).
[8] A. S. P. Gomes, C. R. Jacob, and L. Visscher, Phys. Chem. Chem. Phys. **10**, 5353 (2008).
[9] A. M. Burow, M. Sierka, J. Dobler, and J. Sauer, J. Chem. Phys. **130**, 174710 (2009).
[10] D. Sundholm, J. Chem. Phys. **122**, 194107 (2005).
[11] J. Jusélius and D. Sundholm, J. Chem. Phys. **126**, 094101 (2007).
[12] A. Natan, A. Benjamini, D. Naveh, L. Kronik, M. L. Tiago, S. P. Beckman, and J. R. Chelikowsky, Phys. Rev. B **78**, 075109 (2008).
[13] L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, J. Chem. Phys. **125**, 074105 (2006).
[14] L. Genovese, T. Deutsch, and S. Goedecker, J. Chem. Phys. **127**, 054704 (2007).
[15] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, J. Chem. Phys. **129**, 014109 (2008).
[16] G. Fann, G. Beylkin, R. J. Harrison, and K. E. Jordan, IBM J. Res. Dev. **48**, 161 (2004).
[17] H. Sekino, Y. Maeda, T. Yanai, and R. J. Harrison, J. Chem. Phys. **129**, 034111 (2008).
[18] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, J. Chem. Phys. **121**, 11587 (2004).
[19] R. J. Harrison, G. I. Fann, T. Yanai, and G. Beylkin, Lect. Notes Comput. Sci. **2660**, 103 (2003).
[20] R. J. F. Berger and D. Sundholm, Adv. Quantum Chem. **50**, 235 (2005).
[21] J. Kobus, L. Laaksonen, and D. Sundholm, Comput. Phys. Commun. **98**, 346 (1996).
[22] L. Laaksonen, P. Pyykkö, and D. Sundholm, Int. J. Quantum Chem. **23**, 309 (1983).

I

Paper II

II

A divide and conquer real-space approach for all-electron molecular electrostatic potentials and interaction energies

S. A. Losilla and D. Sundholm
Department of Chemistry, University of Helsinki, FIN-00014 Helsinki, Finland

(Received 9 March 2012; accepted 9 May 2012; published online 4 June 2012)

A computational scheme to perform accurate numerical calculations of electrostatic potentials and interaction energies for molecular systems has been developed and implemented. Molecular electron and energy densities are divided into overlapping atom-centered atomic contributions and a three-dimensional molecular remainder. The steep nuclear cusps are included in the atom-centered functions making the three-dimensional remainder smooth enough to be accurately represented with a tractable amount of grid points. The one-dimensional radial functions of the atom-centered contributions as well as the three-dimensional remainder are expanded using finite element functions. The electrostatic potential is calculated by integrating the Coulomb potential for each separate density contribution, using our tensorial finite element method for the three-dimensional remainder. We also provide algorithms to compute accurate electron-electron and electron-nuclear interactions numerically using the proposed partitioning. The methods have been tested on all-electron densities of 18 reasonable large molecules containing elements up to Zn. The accuracy of the calculated Coulomb interaction energies is in the range of 10^{-3} to 10^{-6} E_h when using an equidistant grid with a step length of 0.05 a_0. © *2012 American Institute of Physics*. [http://dx.doi.org/10.1063/1.4721386]

II

I. INTRODUCTION

All practical implementations of any electronic structure method require representing wave functions and electron densities in a discrete form. Many-body wave functions are typically written as a linear combination of products of single-particle wave functions, implying that the problem is in practice reduced to representing three-dimensional scalar functions. For real-space methods, the functions are expressed as a linear combination of basis functions $\chi_i(\mathbf{r})$,

$$f(\mathbf{r}) \approx \sum_i^N c_i \chi_i(\mathbf{r}). \tag{1}$$

The accuracy of the computation for a given electronic structure model is then limited by the quality of the basis set, that is, how closely the exact solution can be represented by the expansion in Eq. (1). The computational efficiency due to the employed basis sets depends on two factors. First, the compactness of the basis set, that is, how many terms are needed to achieve a given accuracy. Second, the complexity of the computational operations involving the $\chi_i(\mathbf{r})$ functions.

Atom-centered basis sets are the most popular choice in molecular electronic structure calculations. They can be written as

$$\chi_i^A(\mathbf{r}_A) = R_i(r_A) Y_{l_i m_i}(\theta_A, \phi_A), \tag{2}$$

where $R_i(r_A)$ are radial functions and $Y_{l_i m_i}(\theta_A, \phi_A)$ denotes spherical harmonics describing the angular part. The subscript A refers to the nucleus located at position $\mathbf{R}_A$, and r_A, θ_A, ϕ_A are relative spherical coordinates with respect to $\mathbf{R}_A$. The choice of the radial functions $R_i(r)$ defines different families of atom-centered basis sets. The most common are Slater-type (STO; $R_i(r) \propto e^{-\zeta_i r}$) and Gaussian-type (GTO; $R_i(r) \propto e^{-\alpha_i r^2}$) basis sets.[1,2] More flexible numerical forms of $R_i(r)$ have also been employed.[3,4]

The popularity and success of atom-centered basis sets is due to their compactness: even small basis sets such as STO-3G, which consists of 15 Gaussian functions with $l_{max} = 1$ for carbon, can provide qualitatively correct results. To improve the basis set, more radial functions as well as higher angular momentum (l) functions have to be included. Nevertheless, relatively compact basis-set expansions such as the so-called correlation-consistent triple-ζ valence basis sets (cc-pVTZ),[5] which consists of 42 Gaussian functions with $l_{max} = 3$ for C, are considered to provide quantitatively reliable results in most routine calculations. Although calculations with small atom-centered basis sets offer a reasonably good accuracy in spite of their small size, increasing the basis-set size does not necessarily improve the results systematically nor uniformly. A practically complete basis set is difficult to obtain, because of the global nature of the basis functions, that is, their domain spans the whole $\mathbb{R}^3$, hampering the description of local anisotropies. Adding more functions *ad infinitum* is not feasible, because large global basis sets easily become overcomplete, which causes serious numerical problems.

Alternatively, local basis sets can be employed to overcome these issues. Local basis functions are only non-zero inside a given region of space, and they have a very simple mathematical form, such as low-order polynomials. The representation is improved by further subdividing the computational domain to include more local functions. This permits a systematic and consistent convergence with the basis-set size. Certain local basis sets, such as wavelets or multi-resolution adaptive basis sets, can even provide guaranteed precision.[6–8] However, representing the nuclear cusps requires a very large

amount of local functions. Therefore, the common numerical electronic structure programs make use of pseudopotentials to avoid treating the core electrons explicitly.[8–13] All-electron numerical calculations have been traditionally limited to atoms[14,15] and diatomic molecules,[16–20] where it is easy to use grids in special coordinates. Calculations on generic polyatomic molecules are more complicated, as general systems of coordinates are not available. Although attempts to develop fully numerical methodologies have been undertaken for decades,[21,22] numerical schemes have made their way almost exclusively in the integration of the exchange-correlation potential in density functional theory (DFT) codes.[23] Approaches based on finite elements suffer from the difficulty of generating suitable grids.[24] Adaptive schemes, on the other hand, have been able to provide arbitrarily accurate results in Hartree-Fock (HF) or DFT calculations, with energies deviating less than 1 μE_h (micro hartree) from the complete basis-set limit.[6,7,25–27] Nevertheless, an accurate description of the nuclear cusps requires a very large amount of grid points, limiting the calculations to molecules consisting of a small number of atoms.[26]

The question is whether it is possible to devise a numerical basis set that combines the flexibility of local basis sets with the compactness of atom-centered basis sets. Several approaches in this direction can be found in the literature. For periodic systems, plane waves can be employed to describe the smooth electron density of the valence electrons.[28–33] Basis sets composed of both plane waves and spherically symmetric functions are used in several approaches descending from Slater's augmented wave methods.[34–36] Čársky has shown that a mixed Gaussian and plane-wave basis can be used to speed up the calculation of two-electron integrals.[37] For molecular systems, the group of Hirao developed a method for computing electrostatic potentials in a mixed representation of finite elements and Gaussian functions.[38–40] A similar algorithm that is based on a combination of three-dimensional finite elements and numerical spherically symmetric functions has been proposed without any demonstrations of its applicability to molecular systems.[41]

In this work, we show that it is possible to combine atom-centered numerical functions to describe the nuclear cusps with local functions that account for the remaining differences. We propose the following fully numerical representation for molecular scalar three-dimensional functions:

$$f(\mathbf{r}) = \sum_A \sum_{l=0}^{l_M} \sum_{m=-l}^{l} f_{lm}^A(r_A) Y_{lm}(\theta_A, \phi_A) + f^\Delta(\mathbf{r}). \quad (3)$$

In Eq. (3), the atom-centered functions (the *bubbles*) are represented by radial functions $f_{lm}^A(r_A)$ multiplied by the corresponding (real) spherical harmonics $Y_{lm}(\theta_A, \phi_A)$ (Y_{lm}^A in shorthand). The radial functions $f_{lm}^A(r_A)$ are approximated using one-dimensional finite elements. The remainder $f^\Delta(\mathbf{r})$ (the *cube*) is a three-dimensional function, represented using tensorial finite-element functions.

The partitioning into bubbles and cube in Eq. (3) is in principle arbitrary, but it should be chosen such that $f^\Delta(\mathbf{r})$ is cusp-free, smooth, and small-valued compared to the total function $f(\mathbf{r})$.[42] The maximum angular momentum (l_M) required for such a partitioning should be small for keeping the computational complexity tractable. We demonstrate here that molecular electronic densities can be accurately treated using bubbles with $l_M = 1$. This partition offers an additional advantage in electronic structure calculations, namely, the possibility to compute derivatives accurately, by representing all low-order derivative discontinuities at the nuclear positions in the bubbles. This allows an explicit construction of the Hamiltonian, which is not always an option in other real-space approaches.

This paper is structured as follows. The details of the numerical representation are discussed in Sec. II. In Sec. III, we provide algorithms to compute electrostatic potentials, electron-electron and electron-nuclear interaction energies in the bubbles representation. The accuracy of the algorithms is explored for some simple test systems. In Sec. IV, we apply the algorithms to molecular electron densities obtained from DFT calculations using Slater-type orbitals.

II. NUMERICAL FRAMEWORK

A. Numerical representation of one-dimensional functions

The radial parts of the bubbles $f_{lm}^A(r_A)$ are approximated using one-dimensional finite elements, that is, they are expanded as a linear combination of N local basis functions $\{\chi_i(x) : i \in [1, N]\}$ centered at a set of N grid points $X = \{x_i : i \in [1, N]\}$,

$$f(x) \approx \sum_{i=1}^{N} c_i \chi_i(x). \quad (4)$$

The set of grid points is generated by subdividing the domain $[x_{min}, x_{max}]$ into a set of M intervals $\{[a_k, a_{k+1}] : k \in [1, M]\}$, with $a_1 = x_{min}$ and $a_{M+1} = x_{max}$. Each interval contains $P + 1$ equidistant grid points. The last grid point in a given interval is the first of the next one, yielding a total number of grid points of $N = MP + 1$.

The functions $\chi_i(x)$ are Lagrange basis polynomials of order P defined locally in one interval. The ith basis function can be formally defined by means of the *characteristic function* $I_A(x)$ (1 if $x \in A$, and 0 otherwise) as

$$\chi_i(x) = \sum_{j=1}^{M} I_{A_j}(x) I_{A_j}(x_k) \prod_{x_k \in A_j \setminus \{x_i\}} \frac{(x - x_k)}{(x_i - x_k)}. \quad (5)$$

Continuity is ensured, as the basis functions centered at interval boundaries belong to two adjacent intervals. The functions given by Eq. (5) have the property $\chi_i(x_j) = \delta_{ij}$, that is, they are 1 at their corresponding grid point, and 0 at the others. Therefore, the expansion coefficients c_i in Eq. (4) are equal to the

values of the function at the grid points,

$$f(x) \approx \sum_{i=1}^{N} f(x_i)\chi_i(x). \tag{6}$$

The piece-wise nature of the basis in Eq. (5) avoids common problems related to Lagrange interpolation polynomials, in particular due to Runge's phenomenon.

The cusps of the atom-centered functions close to the origin require many grid points close to the origin. In the distant regions, the functions are smooth with small amplitudes, implying that sparser grids suffice. In order to obtain a procedure to generate adequate one-dimensional grids, we studied the representation of $1s$ Slater-type electron densities ($\rho(r) = \pi^{-1}Z^3e^{-2Zr}$), which are the steepest contributions to the electron density. The accuracy criterion was the error in the electrostatic potential, whose analytical form is $V(r) = [1 - e^{-2Zr}(1 + Zr)]r^{-1}$, computed numerically from $\rho(r)$ as discussed in Sec. III. Starting from an equidistant grid with 4 intervals in the range $[0, r_{max}]$, an "optimal" grid was obtained by splitting each interval into two in a recursive manner, until the local contribution to the potential was below a certain threshold.

We observed that the shape of the grids obtained in such way could be approximately reproduced by remapping the boundaries of an equidistant grid with M intervals from 0 to r_{max} using

$$r' = \frac{cr}{c + r_{max} - r}, \tag{7}$$

where $c = 8Z^{-1.5}$, Z being the charge of the nucleus where the bubble is centered at. Numerical tests showed that for a constant number of grid intervals, the error in the potential is proportional to Z^2. On the other hand, for a fixed Z, the error was proportional to M^{-8}. Therefore, in order to have a more or less constant precision, the number of intervals M should be proportional to $Z^{1/4}$. In other words, if M cells are chosen for hydrogen ($Z = 1$), $2M$ are used for sulphur ($Z = 16$), etc. r_{max} was set to 20 a_0 regardless of Z.

The grids obtained with $r_{max} = 20\ a_0$ and $M_{Z=1} = 100$, using 6th order Lagrange interpolation polynomials, enable the present algorithms to provide an adequate accuracy of 9–12 digits in the electrostatic potential. Thus, even for the heaviest elements, a few thousand grid points should be sufficient (e.g., 1795 points for Hg, $Z = 80$). The size of the one-dimensional grids has very little influence on the overall computational performance and the size can be increased when necessary.

B. Numerical representation of three-dimensional functions

The *cubes* are represented by means of *tensorial* finite elements. The grid points are obtained as a Cartesian product of 3 one-dimensional grids,

$$\begin{aligned} P &= X \times Y \times Z \\ &= \{\mathbf{r}_{ijk} = (x_i, y_j, z_k) : i \in [1, N_x],\ j \in [1, N_y],\ k \in [1, N_z]\}. \end{aligned} \tag{8}$$

For each one-dimensional grid, a corresponding basis set is generated as given by Eq. (5). The three-dimensional basis set is a tensor product of the one-dimensional bases,

$$\begin{aligned} &\{\chi_{ijk}(x, y, z)\} \\ &\quad = \{\chi_i^x\} \otimes \{\chi_i^y\} \otimes \{\chi_k^z\} \\ &\quad = \{\chi_i^x(x)\chi_j^y(y)\chi_k^z(z) : i \in [1, N_x],\ j \in [1, N_y],\ k \in [1, N_z]\}. \end{aligned} \tag{9}$$

Similar to the one-dimensional case in Eq. (4), the three-dimensional functions are approximated as

$$f^{\Delta}(x, y, z) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z} f_{ijk}^{\Delta}\chi_i^x(x)\chi_j^y(y)\chi_k^z(z), \tag{10}$$

using the shorthand notation $f_{ijk}^{\Delta} \equiv f^{\Delta}(x_i, y_j, z_k)$.

Currently, a simple equidistant grid is used. More general grids to represent the three-dimensional functions can be implemented without modifying the rest of the code. The three-dimensional domain is defined such that every atom is at least 8 a_0 away from the boundaries. Preliminary tests show that the number of grid points can be reduced by a factor of at least two per dimension without affecting the accuracy when simple non-equidistant grids are employed.

III. ELECTROSTATICS

A. Calculation of electrostatic potentials

The electrostatic potential produced by a charge density partitioned as in Eq. (3) can be computed in a straightforward manner by integrating the Coulomb potential, as each term will yield a potential contribution of the same symmetry

$$\begin{aligned} V(\mathbf{r}) &= \int_{\Omega} \frac{\sum_{A,lm} \rho_{lm}^A(r'_A)Y_{lm}(\theta'_A, \phi'_A) + \rho^{\Delta}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\mathrm{d}^3r' \\ &= \sum_{A,lm} V_{lm}^A(r_A)Y_{lm}^A + V^{\Delta}(\mathbf{r}), \end{aligned} \tag{11}$$

where $Y_{lm}^A = Y_{lm}(\theta_A, \phi_A)$.

The radial functions $V_{lm}^A(r_A)$ can be calculated efficiently and accurately by means of the following expression:

$$\begin{aligned} V_{lm}^A(r_A) = \frac{4\pi}{2l+1}\Bigg[&{r_A}^{-(l+1)} \int_0^{r_A} \rho_{lm}^A(s_A)s_A^{l+2}\mathrm{d}s_A \\ &+ {r_A}^{l} \int_{r_A}^{\infty} \rho_{lm}^A(s_A)s_A^{1-l}\mathrm{d}s_A \Bigg]. \end{aligned} \tag{12}$$

For $1s$ Slater-type densities with Z up to 30, calculations using the one-dimensional grid described in Sec. II A yield potentials accurate to 12 digits. The largest errors of ca. 10^{-9} a.u. appear only very close to the nuclear positions. The computational cost for the integration of the expression in Eq. (12) is negligible as compared to the CPU time spent in the rest of the code.

For the calculation of V^{Δ}, the direct three-dimensional integration method is used.[43–45] It exploits the tensorial

II

finite-element basis in Eq. (10) leading to almost linear ($\mathcal{O}(N^{4/3})$) scaling with the number of grid points (N). Linear scaling can be achieved by using an approach related to the fast multipole method (FMM).[46]

The Coulomb operator is written using the integral expression

$$\frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}=\frac{1}{r_{12}}=\frac{2}{\sqrt{\pi}}\int_0^\infty e^{-t^2r_{12}^2}\mathrm{d}t. \tag{13}$$

The integration domain is split in two parts: the integral from 0 to a given t_f (long range) is performed numerically using Legendre-Gauss quadrature in both linear and logarithmic coordinates. From t_f to infinity (short range) the Gaussian is approximated as a Dirac delta function. The resulting expression is

$$\frac{1}{r_{12}}\approx\sum_p^{p_{max}}\omega_p e^{-t_p^2r_{12}^2}+\frac{\pi}{t_f^2}\delta(\mathbf{r}_1-\mathbf{r}_2), \tag{14}$$

where t_p and ω_p are the quadrature points and the integration weights, respectively. Applying the operator in this form on ρ^Δ and integrating over the $\mathbf{r}_2$ space, we obtain the final expression

$$V_{ijk}^\Delta=\frac{2}{\sqrt{\pi}}\sum_p^{p_{max}}\omega_p\sum_{k'}^{N_z}F_{kk'}^{z,p}\sum_{j'}^{N_y}F_{jj'}^{y,p}\sum_{i'}^{N_x}F_{ii'}^{x,p}\rho_{i'j'k'}^\Delta+\frac{\pi}{t_f^2}\rho_{ijk}^\Delta, \tag{15}$$

where the elements of the $\mathbf{F}^p$ matrices are computed as

$$F_{ii'}^{\xi,p}=\int_{-\infty}^{\infty}e^{-t_p^2(\xi_i-\xi_2)^2}\chi_{i'}^\xi(\xi_2)\mathrm{d}\xi_2 \tag{16}$$

for $\xi=x,y,z$. In Eq. (16), ξ_i and $\chi_i^\xi(\xi)$ are the ith grid point and basis function in direction ξ, respectively.[43–45]

The accuracy can be systematically improved by increasing the number of quadrature points. Sub-μE_h precision for interaction energies can be achieved for Gaussian charge distributions with exponents of $\alpha=0.6$–1.2 when using grid steps of around $0.1\,a_0$, 60 quadrature points, and $t_f=500$. For the calculations presented in this work, we used 20 quadrature points (12 points in the linear range and 8 points in the logarithmic range), with $t_l=2$ and $t_f=500$.

Equation (15) can be reorganized as a series of matrix multiplications. This enables a very fast performance on general-purpose graphics processing units (GPGPUs). A preliminary implementation on an nVidia Tesla C1060 card run at about 22 GFLOPs (data transfer included) for a 361^3 grid, which is 28% of the theoretical peak performance of the device. Although the periodicity has not been considered in the present work, it can be taken into account at virtually no computational costs rendering calculations on a variety of systems such as surfaces, wires, etc., feasible.[45]

B. The energy of the two-body interactions

The electrostatic interaction energy (E_2) between two charge distributions is a key quantity in electronic structure calculations,

$$E_2=\int_\Omega\rho(\mathbf{r})V(\mathbf{r})\,\mathrm{d}^3r. \tag{17}$$

When $V(\mathbf{r})$ and $\rho(\mathbf{r})$ are partitioned as in Eq. (3), E_2 can be computed in a straightforward manner by integrating the product term by term,

$$\begin{aligned}E_2=&\frac{4\pi}{2l+1}\sum_{A,lm}\int_0^\infty\rho_{lm}^A(r_A)V_{lm}^A(r_A)r_A^2\mathrm{d}r_A\\&+\sum_{A\neq B,lm,l'm'}\int_\Omega\rho_{lm}^A(r_A)Y_{lm}^AV_{l'm'}^B(r_B)Y_{l'm'}^B\mathrm{d}^3r\\&+\sum_{A,lm}\int_0^\infty\rho_{lm}^A(r_A)Y_{lm}^AV^\Delta(\mathbf{r})\mathrm{d}^3r\\&+\sum_{A,lm}\int_0^\infty\rho^\Delta(\mathbf{r})V_{lm}^A(r_A)Y_{lm}^A\mathrm{d}^3r+\int_\Omega\rho^\Delta(\mathbf{r})V^\Delta(\mathbf{r})\mathrm{d}^3r.\end{aligned} \tag{18}$$

Such an approach is not recommended because of two reasons. First, the number of cross-terms grows with the square of the number of centers. Second, and more importantly, only the first terms, which are products of spherical functions centered at the same atom, retain spherical symmetry and can be easily computed. The 3rd and 4th terms are particularly difficult to integrate, because they are the product of a bubble and a cube, therefore they are general three-dimensional functions with steep cusps.

Both problems can be circumvented by partitioning the integrand, that is, the energy density in Eq. (17) into *bubbles* and *cube* parts,

$$\rho(\mathbf{r})V(\mathbf{r})=U(\mathbf{r})=\sum_{Alm}U_{lm}^AY_{lm}^A+U^\Delta. \tag{19}$$

A function partitioned in this way is very easy to integrate, as only the spherically symmetric U_{00}^A terms survive due to the symmetry of the spherical harmonics,

$$E_2=4\pi\sum_A\int_0^\infty U_A^{00}(r_A)r_A^2\mathrm{d}r_A+\int_\Omega U^\Delta(\mathbf{r})\mathrm{d}^3r. \tag{20}$$

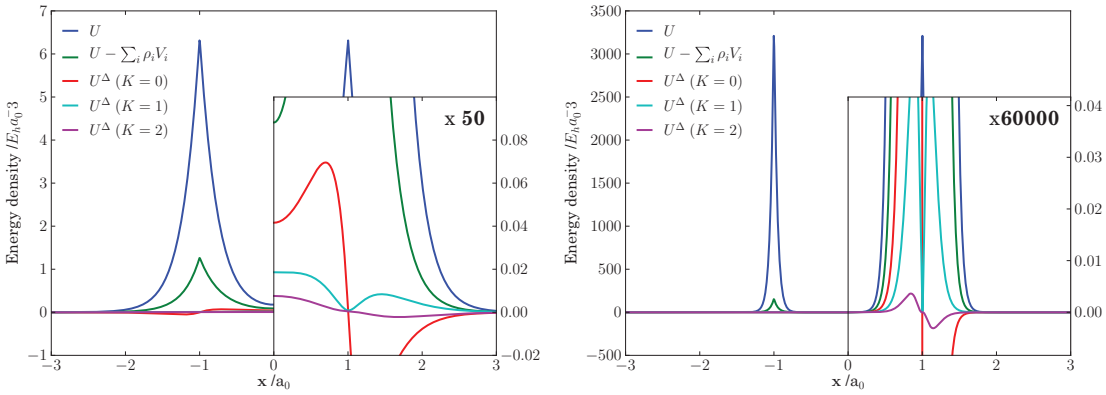
The accuracy of the integration relies on how small and smooth $U^\Delta(\mathbf{r})$ is, which in turn depends on the construction of adequate $U_{lm}^A(r_A)$ functions.

We can express $U(\mathbf{r})$ in the form of Eq. (19) as

$$\begin{aligned}U(\mathbf{r})=&\sum_{A,lm}\left[\rho_{lm}^AY_{lm}^A+\rho^\Delta(\mathbf{r})\right]V(\mathbf{r})\\\approx&\sum_{A,l_1m_1,l_2m_2}\rho_{l_1m_1}^AY_{l_1m_1}^A\tilde{V}_{l_2m_2}^AY_{l_2m_2}^A+\rho^\Delta(\mathbf{r})V(\mathbf{r}).\end{aligned} \tag{21}$$

The relation in Eq. (21) is exact if the potential can be expressed as a one-center expansion,

$$V(\mathbf{r})=\sum_{lm}\tilde{V}_{lm}^A(r_A)Y_{lm}^A. \tag{22}$$



(a) $Z = 2$ (b) $Z = 10$

FIG. 1. U^Δ (in $E_h a_0^{-3}$) along the x axis for a superposition of two $1s$ charge densities of hydrogen-like atoms with $Z = 2$ and 10 at $x = \pm 1\ a_0$, after subtracting bubbles generated with different Taylor series orders K. The right hand sides of the graphs are magnified to show $U^\Delta(K = 2)$.

The $U^A_{lm}(r_A)$ functions can then be obtained by projection such that

$$U^A_{lm}(r_A) = \sum_{l_1 m_1, l_2 m_2} \langle Y_{lm} | Y_{l_1 m_1} Y_{l_2 m_2} \rangle \rho^A_{l_1 m_1}(r_A) \tilde{V}^A_{l_2 m_2}(r_A). \quad (23)$$

The remaining question is how to obtain the radial parts $\tilde{V}^A_{lm}(r_A)$. The obvious approach is some kind of fitting algorithm, which would though be too computationally expensive. Instead, the known information about $\rho^A_{lm}(r_A)$ and $V^A_{lm}(r_A)$ can be exploited to construct the $U^A_{lm}(r_A)$ functions.

Let us denote the parts of $V(\mathbf{r})$ that are not centered at $\mathbf{R}_A$ as

$$V^{\neq}_A(\mathbf{r}) = V(\mathbf{r}) - \sum_{lm} V^A_{lm} Y^A_{lm}. \quad (24)$$

At nuclear positions $\mathbf{R}_B \neq \mathbf{R}_A$, $V^{\neq}_A(\mathbf{R}_A)$ is smooth, and can be approximated as a series expansion to order K. The resulting terms can be reorganized to have the same symmetry as the spherical harmonics. The resulting expression for $\tilde{V}^A_{lm}(r_A)$ is thus

$$\tilde{V}^A_{lm}(r_A) = V^A_{lm}(r_A) + \sum_{|\boldsymbol{\kappa}| \leq K} C^{\boldsymbol{\kappa}}_{lm} \frac{(\partial^{\boldsymbol{\kappa}} V^{\neq}_A)(\mathbf{R}_A)}{\boldsymbol{\kappa}!} r_A^{|\boldsymbol{\kappa}|}, \quad (25)$$

where $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3)$ is a three-dimensional multi-index, with $\partial^{\boldsymbol{\kappa}} = \partial_x^{\kappa_1} \partial_y^{\kappa_2} \partial_z^{\kappa_3}$, $|\boldsymbol{\kappa}| = \kappa_1 + \kappa_2 + \kappa_3$, and $\boldsymbol{\kappa}! = \kappa_1! \kappa_2! \kappa_3!$. The $C^{\boldsymbol{\kappa}}_{lm}$ coefficients are given by

$$C^{\boldsymbol{\kappa}}_{lm} = \int_0^{2\pi} \int_0^{\pi} \frac{\mathbf{r}^{\boldsymbol{\kappa}}_A}{r_A^{|\boldsymbol{\kappa}|}} Y_{lm}(\theta_A, \phi_A) \sin(\theta_A) \mathrm{d}\phi_A \mathrm{d}\theta_A, \quad (26)$$

with $\mathbf{r}^{\boldsymbol{\kappa}}_A = x_A^{\kappa_1} y_A^{\kappa_2} z_A^{\kappa_3}$. The remainder U^Δ can be calculated at each grid point by subtracting the $U^A_{lm} Y^A_{lm}$ bubbles from the total U. One can see that when $K \to \infty$ in Eq. (25), Eq. (22) holds exactly, and thus from Eqs. (19) and (21) we can identify $U^\Delta = \rho^\Delta V$.

Figure 1 shows U^Δ for different K values obtained for a superposition of two $1s$ charge densities of hydrogen-like atoms at a distance of 2 a_0 from each other. Although the diagonal terms are the dominant contribution, the terms arising from $V^{\neq}_A(\mathbf{r})$ must be included to obtain a smooth U^Δ. At least the first term in the series expansion ($K = 0$) should be included to yield a cusp-free U^Δ. Each additional order of K included in the series expansion renders a smaller U^Δ, which becomes crucial in order to have U^Δ of the same order of magnitude regardless of the atomic charge, as seen when comparing U^Δ for $K = 1$ in Figures 1(a) and 1(b).

Formally, the bubbles algorithm for the two-body interaction energy scales as $\mathcal{O}((K+1)^2 N_{at}^2)$, when calculating the Taylor series expansion of every spherical function at every other center. In practice, the main bottleneck is interpolating all radial functions at all grid points, which scales linearly with both the number of grid points and atoms ($\mathcal{O}(3(\max\{l_M, K\})^2 N_{at} N_g)$). However, this task can be parallelized very efficiently, particularly on GPGPUs.

In Figure 2, the error in the electron-electron interaction energy for the model diatomic system is shown for $K = 2$.

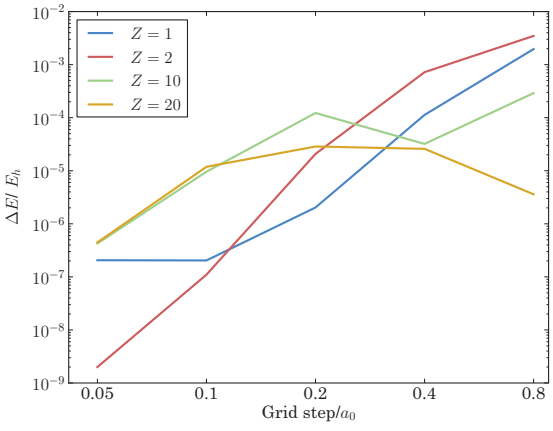


FIG. 2. Errors in the self-interaction energies (in E_h) for superposition of two hydrogen-like $1s$ charge densities at $x = \pm 1\ a_0$, for different values of Z.

The errors decrease in a similar manner for all Z values when a denser grid is used. However, the error increases for the larger Z values. The reason is the small-scale detailed features in U^{Δ}, which require small grid steps to be described accurately (see Figure 1(b)). For the same reason, error cancellations might appear for larger Z values. For $K = 2$, the integral of these oscillating regions is almost vanishing. When a coarse grid is used, the oscillations are not represented, integrating to exactly 0. Nevertheless, for all studied Z values, the accuracy could be made smaller than 10^{-6} E_{h} using the densest grids.

C. The energy of the one-body interactions

The electrostatic interaction energy between a set of nuclei or other point charges moving in the potential arising from the electronic charge density (E_{Nuc}) is another crucial term to be computed in molecular electronic structure calculations. It can be written as

$$E_{Nuc} = \sum_{A} -Z_A V(\mathbf{R}_A), \tag{27}$$

where Z_A denotes the charge of nucleus A. This can be trivially computed once the electrostatic potential caused by the electron density $V(\mathbf{r})$ has been computed, as described in Sec. III B. The accuracy will be limited by the accuracy of the potential at the nuclear positions. As discussed previously, the contributions from the bubbles can be obtained with a very high accuracy at a small cost. Hence, the limiting factor in the accuracy is the contribution from the three-dimensional remainder. For cases where $\rho^{\Delta} = 0$, the accuracy is only limited by the one-dimensional grid. For the superposition of hydrogen-like $1s$ charge distribution discussed in Sec. III B, the error in E_{Nuc} is smaller than $10^{-6}E_{\mathrm{h}}$ for $Z \leq 20$.

If the nuclei are not treated as point charges but have a finite size, Eq. (27) cannot be used. The problem is then equivalent to computing the electrostatic interaction energy between two charge distributions, as discussed in Sec. III B. The additional difficulty lies on the steepness of the potential close to the nuclear centers. In such case, we can partition the energy density,

$$W(\mathbf{r}) = \zeta(\mathbf{r})\rho(\mathbf{r}) = \sum_{A} \zeta_A(r_A)\rho(\mathbf{r}). \tag{28}$$

In Eq. (28), we have assumed spherically symmetric nuclei for simplicity. The corresponding radial functions $W^A_{lm}(r_A)$ can be constructed as

$$W^A_{lm}(r_A) = \zeta_A(r_A)\left[\rho^A_{lm}(r_A) + \alpha^A_{lm}(r_A)\right] + \beta^A_{lm}(r_A). \tag{29}$$

The function $\alpha^A_{lm}(r_A)$ is given by

$$\alpha^A_{lm}(r_A) = \mu(r_A) \sum_{|\kappa|\leq K} C^{\kappa}_{lm} \frac{(\partial_{\kappa}\rho^{\neq}_A)(\mathbf{R}_A)}{\kappa!} r_A^{|\kappa|}. \tag{30}$$

The *fuzzy* function $\mu(r)$ ensures that the contributions arising from the series expansion are forced to decay to 0, to avoid that the term $\zeta_A(r_A)\alpha^A_{lm}(r_A)$ diverges at long distances. A possible $\mu(r)$ that satisfies this condition is

$$\mu(r) = \frac{1}{2}\mathrm{erfc}\left(\frac{r^2 - 1}{r}\right). \tag{31}$$

$\mu(r)$ is flat at $r = 0$, that is, $\frac{\partial^n \mu(r)}{\partial r^n}|_{r=0} = 0 \,\forall\, n > 0$, implying that $\mu(r)$ can be considered equal to 1 when r is close to 0. $\beta^A_{lm}(r_A)$ are radial functions, which are defined as

$$\beta^A_{lm}(r_A) = \sum_{l_1 m_1, l_2 m_2} \langle Y_{lm}|Y_{l_1 m_1} Y_{l_2 m_2}\rangle \rho^A_{l_1 m_1}(r_A)\tilde{\zeta}^A_{l_2 m_2}(r_A) \tag{32}$$

with

$$\tilde{\zeta}^A_{lm}(r_A) = \sum_{|\kappa|\leq K} C^{\kappa}_{lm} \frac{(\partial_{\kappa}\zeta^{\neq}_A)(\mathbf{R}_A)}{\kappa!} r_A^{|\kappa|}. \tag{33}$$

The resulting $W^A_{lm}(r_A)$ functions are very steep, although they can be easily integrated in spherical coordinates after multiplying by the volume element $4\pi r^2$. In order to avoid numerical instabilities, the remainder W^{Δ} can be computed directly as

$$W^{\Delta}(\mathbf{r}) = \sum_{A} \left\{ \zeta_A(r_A) \left[\rho^{\Delta}(\mathbf{r}) + \sum_{lm} \left(\sum_{B\neq A} \rho^B_{lm}(r_B) Y^B_{lm} - \alpha^A_{lm}(r_A) Y^A_{lm} \right) \right] - \sum_{lm} \beta^A_{lm}(r_A) Y^A_{lm} \right\}. \tag{34}$$

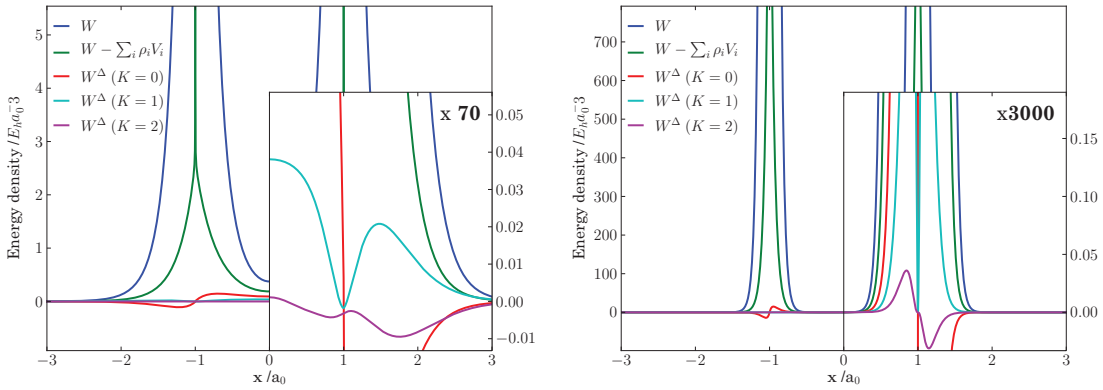
Notice that the term in the square brackets in Eq. (34) is zero at $r_A = 0$.

This procedure is obviously much more expensive than the evaluation of Eq. (27). The scaling of the algorithm is the same as for the self-interaction energy, although the prefactor is larger as more functions have to be interpolated.

Figure 3 shows the obtained W^{Δ} functions for a diatomic test system using different K values, for the case of point-like nuclei. The shape of the W^{Δ} functions is similar to the analogously obtained U^{Δ} functions. The accuracy of the integrated interaction energies is approximately 2 orders of magnitude lower than the corresponding electron-electron repulsion energies for the test system, which is apparent by comparing the graphs in Figures 1(b) and 3(b).

IV. APPLICATION TO MOLECULAR ELECTRON DENSITIES

In this section, the accuracy of the presented algorithms is assessed by computing the total Coulomb energy, that is, the self-interaction plus nuclear-electron interaction energies of molecular electron densities obtained from electronic structure calculations using STOs. Thus, we explore not only the accuracy of the method itself, but also the possibility of partitioning realistic electron densities into cubes and bubbles with $l_M = 1$, that is, using spherical functions of s and p symmetries for the bubbles.



(a) $Z = 2$ (b) $Z = 10$

FIG. 3. W^Δ (in $E_h a_0^{-3}$) along the x axis for a superposition of two hydrogenoid $1s$ charge densities with $Z = 2$ and 10 at $x = \pm 1\ a_0$, after subtracting bubbles generated with different Taylor series orders K. The right hand sides of the graphs are magnified to show $W^\Delta(K = 2)$.

A. A partitioning scheme for molecular electron densities

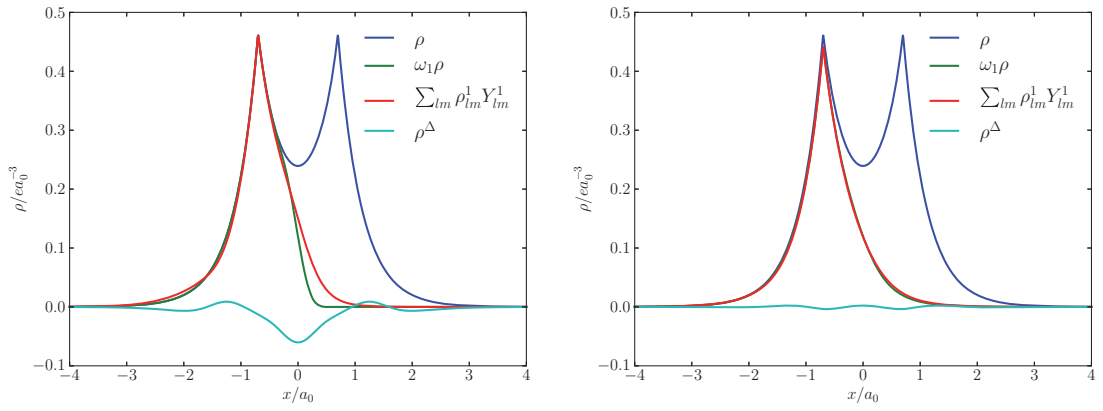
An arbitrary density cannot be partitioned into bubbles and cube by using a straightforward projection, as the bubbles overlap, that is, the basis is not orthonormal. Alternatively, the cube can be computed as a remainder once the radial functions of the bubbles are determined. Appropriate bubbles can be obtained by dividing the total density into atomic domains, which are projected onto the corresponding spherical harmonics basis,

$$\rho^A_{lm}(r_A) = \int_0^{2\pi}\int_0^{\pi} \omega_A(\mathbf{r})\rho(\mathbf{r})Y_{lm}(\theta_A, \phi_A)\sin(\theta_A)\mathrm{d}\theta_A\mathrm{d}\phi_A. \tag{35}$$

In Eq. (35), the division into subdomains is achieved by multiplying the total density ρ with a *masking function* ω_A. Following the approach of Becke,[47] we require that $\sum_A \omega_A(\mathbf{r}) = 1\ \forall\, \mathbf{r} \in \mathbb{R}^3$. The form of $\omega_A(\mathbf{r})$ suggested by Becke is not suitable in this case, because the decay rate is not exponential which yields very asymmetric atomic contributions. Thus, the resulting radial functions calculated using Eq. (35) perform poorly in the inter-atomic region. This is illustrated in Figure 4(a) for H_2. Alternatively, the ω_A masking functions can be constructed in an iterative manner as

$$\omega_A^{(n+1)}(\mathbf{r}) = \frac{\rho_{00}^{A(n)}(r_A)}{\sum_B \rho_{00}^{B(n)}(r_B)}. \tag{36}$$

In Eq. (36), only the $l = 0$ bubbles are used to construct the masking functions, in order to avoid situations where $\sum_B \rho_{00}^{B(n)}(r_B) = 0$, which leads to numerical problems. Non-iterative versions of Eq. (36) have been used by several other partition schemes, as in the aforementioned one by Becke, and in Hirshfeld partitioning,[48] where ρ_{00}^A are the spherically averaged free-atom electron densities. The Hirshfeld approach does not provide adequate partitioning when hydrogen atoms are present, because the density around the hydrogen nucleus varies wildly depending on the bonding situation.



(a) Becke (b) Iterative

FIG. 4. Comparison of the bubbles obtained with (a) Becke's[47] and (b) the iterative partitioning scheme presented in this work. The total density corresponds to H_2 computed using LDA with DZ basis in ADF.

The angular integration in Eq. (35) is carried out numerically, by sampling at 6 grid points forming an octahedron for each value of r_A. Convergence over all space cannot be expected, thus the iterative procedure is repeated until ρ^{Δ} is smooth enough. This method is very sensitive to the initial choice of radial functions. Several approaches have been tested, and using $\rho_{00}^{B(0)} = e^{-2r}$ for every element as initial guess turned out very successful, also for heteroatomic systems. The cusp densities converge quickly and render smooth remainders close to the nuclear positions. The iterative procedure does not converge over the whole domain. We therefore fixed the number of iterations to 4. This is of course a completely *ad hoc* solution, whose only aim is to provide practical input data that can be used as a realistic test case for our method.

II

B. Test cases

The partition scheme was applied to a set of 18 molecules which was selected to contain a variety of elements up to the fourth row (H, C, O, N, F, S, Zn), to present different bonding situations, and to consist of a reasonably large number of atoms (up to 15). The molecular structures were obtained using the in-built molecular mechanics (MM) pre-optimizer in the ADFinput utility of the ADF program suite.[49–51] The electron densities were obtained at the local density approximation (LDA), with a double-ζ STO basis set and a large frozen core, also using ADF. The electron densities are appropriate for the purpose of testing the accuracy of the methods. For the numerical integration, an accuracy of 6 digits was requested. The interface to read the ADF densities prior to partitioning was done using DGrid.[52]

In Figure 5, the error in the integrated number of electrons ($N_e = \int \rho(\mathbf{r})\mathrm{d}^3r$) is plotted as a function of the grid step. The behavior is similar for all investigated molecules.

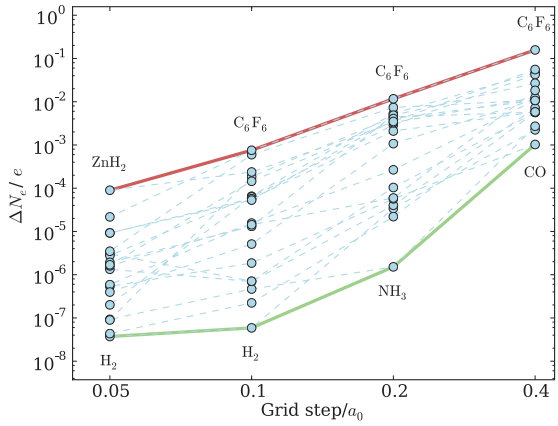


FIG. 5. Error in the number of electrons ($N_e = \int \rho(\mathbf{r})\mathrm{d}^3r$) with respect to the grid step. The trends for each individual molecule are represented with dashed lines. The wider lines represent the largest and smallest errors for each grid step. The molecule yielding the largest and smallest errors is also given for each grid step. The specific values are given in Table I.

The error decreases approximately by a factor of 20 when the grid step is halved. The accuracy is very irregular, with the errors spanning over 3 orders of magnitude for a given grid step. This can be attributed to the partition scheme, as it does not correlate with the number of electrons nor with the atomic numbers.

We report errors in the total Coulomb energy, which is the sum of the interaction energy between the electron density and the nuclear charges plus the electron-electron self-interaction energy. The self-interaction energy is given by Eq. (17), where V is the potential arising from ρ itself, with a factor of $1/2$ to avoid double counting. Errors for the separate

TABLE I. Error in the total number of electrons with respect to the grid step.

		Step/a_0			
Molecule	N_e	0.4	0.2	0.1	0.05
H_2	2	2.2×10^{-3}	-3.4×10^{-5}	5.9×10^{-8}	3.7×10^{-8}
CH_4	10	-7.2×10^{-3}	4.0×10^{-5}	-7.0×10^{-7}	-1.3×10^{-6}
NH_3	10	-2.7×10^{-3}	-1.5×10^{-6}	4.7×10^{-7}	-9.3×10^{-8}
H_2O	10	5.7×10^{-3}	-2.2×10^{-5}	2.2×10^{-7}	4.3×10^{-8}
CO	14	-1.0×10^{-3}	-5.7×10^{-5}	-1.9×10^{-6}	-5.4×10^{-7}
C_2H_6	18	-5.6×10^{-3}	-1.0×10^{-4}	7.1×10^{-7}	-3.0×10^{-6}
CH_3OH	18	4.7×10^{-2}	-5.3×10^{-3}	-6.5×10^{-5}	-2.8×10^{-6}
H_2O_2	18	1.1×10^{-2}	7.4×10^{-3}	-6.0×10^{-4}	3.5×10^{-6}
CO_2	22	1.1×10^{-2}	2.7×10^{-4}	-1.6×10^{-5}	-5.9×10^{-7}
HCOOH	24	4.2×10^{-2}	2.9×10^{-3}	6.3×10^{-5}	-2.0×10^{-7}
O_3	24	-5.7×10^{-3}	-4.5×10^{-3}	1.8×10^{-4}	9.0×10^{-8}
C_3H_8	26	1.1×10^{-2}	-1.1×10^{-3}	5.1×10^{-6}	4.0×10^{-7}
C_2H_5OH	26	-2.7×10^{-2}	-4.9×10^{-3}	1.3×10^{-5}	-1.9×10^{-6}
CH_3COOH	32	5.8×10^{-3}	-2.1×10^{-3}	5.7×10^{-5}	-9.3×10^{-6}
ZnH_2	32	1.2×10^{-2}	3.3×10^{-3}	-2.4×10^{-4}	-9.0×10^{-5}
C_6H_6	42	-1.8×10^{-2}	-5.9×10^{-5}	1.4×10^{-5}	-1.6×10^{-6}
H_2SO_4	50	1.1×10^{-2}	4.1×10^{-3}	-5.3×10^{-5}	9.2×10^{-6}
Adenine	70	5.6×10^{-2}	3.5×10^{-3}	1.4×10^{-4}	1.7×10^{-6}
C_6F_6	90	1.6×10^{-1}	1.2×10^{-2}	-7.5×10^{-4}	-2.2×10^{-5}

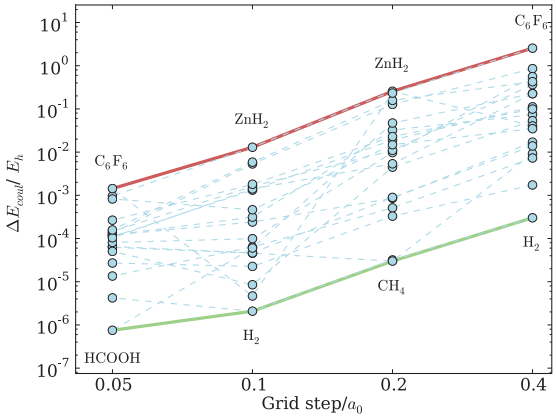


FIG. 6. Error in the total Coulomb energy (in E_h) with respect to the grid step. The trends for each individual molecule are represented with dashed lines. The wider lines represent the largest and smallest errors for each grid step. The molecule yielding the largest and smallest errors is also given for each grid step. The specific values are given in Table II.

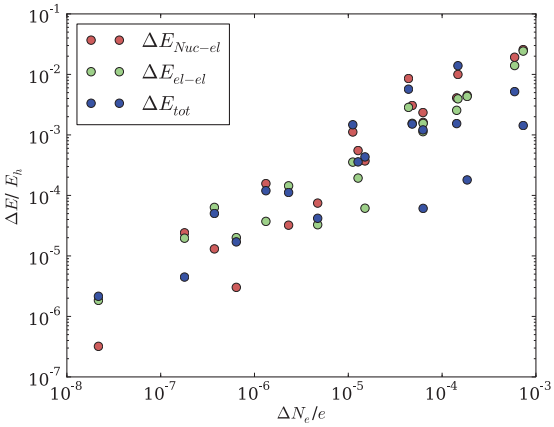


FIG. 7. Difference between the Coulomb energies obtained using the 0.1 and 0.05 a_0 grids, plotted against the difference in the numbers of electrons obtained with those same grids. Each point corresponds to one of the molecules discussed in Sec. IV.

II

contributions cannot be reported, because ADF does not provide separate values for the electron-electron repulsion and the nuclear-electronic attraction energy values, and electrostatic potentials can only be computed at given grids. Figure 6 summarizes these results.

The main bottleneck in the accuracy is the precision of the representation of the residual electron density $\rho^{\Delta}(\mathbf{r})$, that is, its smoothness. In this case, it is mostly dependent on the partitioning scheme chosen. One way to measure the smoothness of $\rho^{\Delta}(\mathbf{r})$ is the change in N_e with the length of the grid step; if $\rho^{\Delta}(\mathbf{r})$ is smooth, N_e will change little when a finer grid is used. In Figure 7, the difference in the Coulomb energies computed with the 0.1 and 0.05 a_0 grids is represented against the difference in the number of electrons obtained with the same grids. It can be seen that the rate at which these quantities converge is closely related. Both the electron-electron and electron-nuclear interaction energies converge in a similar manner. It should be noted that there is no correlation between the error in the energy and the total residual charge ($N_e^{\Delta} = \int \rho^{\Delta}(\mathbf{r})\mathrm{d}^3r$). Hence, the accuracy is not limited by the size of the system, but on how efficiently the total charge density can be partitioned into bubbles and cube.

TABLE II. Error in the total Coulomb energy (in E_h) with respect to the grid step.

		Step/a_0			
Molecule	Energy	0.4	0.2	0.1	0.05
H_2	−2.307736	3.0×10^{-4}	-3.2×10^{-5}	-2.1×10^{-6}	-4.2×10^{-6}
CH_4	−87.206333	9.3×10^{-3}	-3.0×10^{-5}	-4.6×10^{-5}	-6.3×10^{-5}
NH_3	−116.408356	-7.3×10^{-3}	5.1×10^{-4}	6.4×10^{-5}	1.4×10^{-5}
H_2O	−151.941561	1.7×10^{-3}	3.3×10^{-4}	2.3×10^{-5}	2.7×10^{-5}
C_2H_6	−188.072104	-1.4×10^{-2}	-8.9×10^{-4}	4.7×10^{-5}	-6.5×10^{-5}
CO	−228.979498	-1.7×10^{-2}	8.6×10^{-4}	2.4×10^{-4}	1.2×10^{-4}
CH_3OH	−255.122380	-1.1×10^{-1}	1.9×10^{-2}	1.3×10^{-3}	1.1×10^{-4}
C_3H_8	−300.583289	-5.3×10^{-2}	4.5×10^{-3}	8.5×10^{-6}	5.0×10^{-5}
H_2O_2	−319.910606	8.6×10^{-1}	-1.3×10^{-1}	5.3×10^{-3}	1.3×10^{-4}
C_2H_5OH	−368.262932	3.7×10^{-1}	-1.1×10^{-2}	1.6×10^{-3}	1.5×10^{-4}
CO_2	−408.489901	-7.4×10^{-2}	-3.2×10^{-2}	-3.1×10^{-4}	1.2×10^{-4}
HCOOH	−422.585196	2.3×10^{-1}	-4.7×10^{-2}	-6.0×10^{-5}	7.5×10^{-7}
O_3	−492.185520	-5.6×10^{-1}	9.8×10^{-3}	-9.9×10^{-5}	8.1×10^{-5}
CH_3COOH	−544.456301	-2.3×10^{-1}	-5.4×10^{-3}	-1.4×10^{-3}	1.0×10^{-4}
C_6H_6	−630.330039	3.5×10^{-2}	1.5×10^{-2}	4.6×10^{-4}	8.2×10^{-4}
Adenine	−1354.056414	-9.9×10^{-2}	2.3×10^{-2}	-1.8×10^{-3}	-2.7×10^{-4}
H_2SO_4	−1592.319778	4.3×10^{-1}	-1.6×10^{-1}	5.9×10^{-3}	1.6×10^{-4}
C_6F_6	−2264.439035	-2.5×10^{0}	-2.3×10^{-1}	-4.7×10^{-6}	-1.4×10^{-3}
ZnH_2	−3505.481680	4.0×10^{-2}	-2.6×10^{-1}	-1.3×10^{-2}	9.6×10^{-4}

V. CONCLUSIONS AND OUTLOOK

This work is a first step in the direction towards all-electron fully numerical Kohn-Sham DFT and Hartree-Fock calculations on general molecules. A numerical framework has been presented that can be used to accurately represent orbitals and electron densities. We have developed novel algorithms, which can be employed in basic operations of electronic structure calculations, such as for computing electrostatic potentials and interaction energies. The algorithms have been applied to model systems as well as to all-electron densities of molecules. The obtained accuracy of the calculations suggests that the present numerical algorithms can be used as building blocks for computational methods in more complex electronic structure codes.

In the development of the algorithms, the computing efficiency, scaling, and parallelizability have been kept in mind. The most relevant feature of the presented method is the small amount of grid points required to represent steep cusps with very good accuracy. On the other hand, the simplicity of the three-dimensional numerical grids used permits an easy and efficient parallelization on SIMD (single instruction, multiple data) computer architectures, such as GPGPUs. Porting the main bottlenecks of the algorithm into a CUDA version is underway, showing very promising speedups.

ACKNOWLEDGMENTS

The authors would like to thank Jonas Jusélius, and Dou Du for support with the development and maintenance of the code, Mooses Mehine for providing the ADF calculations, and Mathias Pabst and Mark Watson for useful discussions and suggestions, as well as CSC for providing computational resources, and the Academy of Finland for funding.

[1] J. C. Slater, Phys. Rev. **36**, 57 (1930).
[2] S. F. Boys, Proc. R. Soc. London **200**, 542 (1950).
[3] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Comput. Phys. Commun. **180**, 2175 (2009).
[4] B. Delley, J. Chem. Phys. **92**, 508 (1990).
[5] T. H. Dunning, J. Chem. Phys. **90**, 1007 (1989).
[6] T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, J. Chem. Phys. **121**, 6680 (2004).
[7] T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, J. Chem. Phys. **121**, 2866 (2004).
[8] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, J. Chem. Phys. **129**, 014109 (2008).
[9] A. Natan, A. Benjamini, D. Naveh, L. Kronik, M. L. Tiago, S. P. Beckman, and J. R. Chelikowsky, Phys. Rev. B **78**, 75109 (2008).
[10] A. Castro, H. Appel, and M. Oliveira, Phys. Status Solidi B **243**, 2465 (2006).
[11] A. K. Roy, Chem. Phys. Lett. **461**, 142 (2008).
[12] T. Torsti, T. Eirola, J. Enkovaara, T. Hakala, P. Havu, V. Havu, T. Höynälänmaa, J. Ignatius, M. Lyly, I. Makkonen, T. T. Rantala, J. Ruokolainen, K. Ruotsalainen, E. Räsänen, H. Saarikoski, and M. J. Puska, Phys. Stat. Sol. B **243**, 1016 (2006).
[13] J. Fang, X. Gao, and A. Zhou, J. Comp. Phys. **231**, 3166 (2012).
[14] D. Sundholm and J. Olsen, J. Chem. Phys. **94**, 5051 (1991).
[15] C. Froese Fischer, Comput. Phys. Commun. **4**, 107 (1972).
[16] E. A. McCullough, Jr., Chem. Phys. Lett. **24**, 55 (1974).
[17] E. A. McCullough, Jr., J. Chem. Phys. **62**, 3991 (1975).
[18] L. Laaksonen, P. Pyykkö, and D. Sundholm, Int. J. Quant. Chem. **23**, 309 (1983).
[19] L. Laaksonen, P. Pyykkö, and D. Sundholm, Int. J. Quant. Chem. **23**, 319 (1983).
[20] J. Kobus, L. Laaksonen, and D. Sundholm, Comput. Phys. Commun. **98**, 346 (1996).
[21] A. D. Becke and R. M. Dickson, J. Chem. Phys. **89**, 2993 (1988).
[22] A. D. Becke and R. M. Dickson, J. Chem. Phys. **92**, 3610 (1990).
[23] G. te Velde and E. J. Baerends, J. Comp. Phys. **99**, 84 (1992).
[24] R. Alizadegan, K. J. Hsia, and T. J. Martinez, J. Chem. Phys. **132**, 034101 (2010).
[25] G. Fann, G. Beylkin, R. J. Harrison, and K. E. Jordan, IBM J. Res. Dev. **48**, 161 (2004).
[26] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, J. Chem. Phys. **121**, 11587 (2004).
[27] F. A. Bischoff and E. F. Valeev, J. Chem. Phys. **134**, 104104 (2011).
[28] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).
[29] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[30] X. Gonze, B. Amadon, P. M. Anglade, J. M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Côté, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. J. T. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G. M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M. J. Verstraete, G. Zerah, and J. Zwanziger, Comput. Phys. Commun. **180**, 2582 (2009).
[31] H. Limbach, A. Arnold, B. Mann, and C. Holm, Comput. Phys. Commun. **174**, 704 (2006).
[32] M. Preuss, W. G. Schmidt, K. Seino, J. Furthmüller, and F. Bechstedt, J. Comp. Chem. **25**, 112 (2004).
[33] P. Pulay, S. Saebo, M. Malagoli, and J. Baker, J. Comp. Chem. **26**, 599 (2005).
[34] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).
[35] K. Schwarz, P. Blaha, and G. K. H. Madsen, Comput. Phys. Commun. **147**, 71 (2002).
[36] G. Lippert, J. Hutter, and M. Parrinello, Mol. Phys. **92**, 477 (1997).
[37] P. Čársky, Int. J. Quant. Chem. **107**, 56 (2007).
[38] Y. Kurashige, T. Nakajima, and K. Hirao, J. Chem. Phys. **126**, 144106 (2007).
[39] M. A. Watson and K. Hirao, J. Chem. Phys. **129**, 184107 (2008).
[40] M. A. Watson, Y. Kurashige, T. Nakajima, and K. Hirao, J. Chem. Phys. **128**, 054105 (2008).
[41] Z. Romanowski and A. F. Jalbout, Acta Phys. Polonica **39**, 1455 (2008).
[42] D. Sundholm, P. Pyykkö, and L. Laaksonen, Mol. Phys. **55**, 627 (1985).
[43] D. Sundholm, J. Chem. Phys. **122**, 194107 (2005).
[44] J. Jusélius and D. Sundholm, J. Chem. Phys. **126**, 94101 (2007).
[45] S. A. Losilla, D. Sundholm, and J. Jusélius, J. Chem. Phys. **132**, 024102 (2010).
[46] C. A. White and M. Head-Gordon, J. Chem. Phys. **101**, 6593 (1994).
[47] A. Becke, J. Chem. Phys. **88**, 2547 (1988).
[48] F. L. Hirshfeld, Theor. Chim. Acta **44**, 129 (1977).
[49] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler, J. Comp. Chem. **22**, 931 (2001).
[50] C. Fonseca Guerra, J. G. Snijders, G. te Velde, and E. J. Baerends, Theor. Chem. Acc. **99**, 391 (1998).
[51] ADF2010, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, 2010.
[52] M. Kohout, Dgrid, version 4.5, Radebeul, 2009.

Paper V

GPGPU-accelerated real-space tools for molecular electronic structure calculations

Sergio A. Losilla

Department of Chemistry, University of Helsinki, P.O. Box 55, FIN-00014 Helsinki, Finland

Mark A. Watson

Department of Chemistry, Frick Chemistry Laboratory, Princeton University, NJ 08544-1013, USA

Alán Aspuru-Guzik

Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138 USA

Dage Sundholm[a)]

Department of Chemistry, University of Helsinki, P.O. Box 55, FIN-00014 Helsinki, Finland

(Dated: 25 October 2013)

We present a GPGPU implementation of the most expensive operations required to perform SCF calculations using the fully numerical, real-space *bubbles* basis. The optimized operations are linear transformations, such as convolutions with the Coulomb kernel, and products of functions. We show that the port to the GPGPU architecture is straightforward, requiring no substantial modification of the original algorithms. The high performance obtained in the benchmarks is the proof of the aptness of the presented methods for emerging many-core computer architectures. We also present timings for the calculation of electrostatic potentials and interaction energies for all-electron electronic densities for molecules including up to 15 atoms.

[a)]Electronic mail: sundholm@chem.helsinki.fi

I. INTRODUCTION

Molecular electronic-structure calculations are mainly performed using methods that employ atomic orbitals (AOs), which are expanded in basis functions centered at the nuclear positions. The atom-centered basis functions can be expressed in spherical coordinates as

$$\chi_i(\mathbf{r}) = \sum_{l=0}^{L} \sum_{m=-l}^{l} R_{Alm}(r_A) Y_{lm}(\theta_A, \phi_A), \tag{1}$$

where r_A, θ_A and ϕ_A are the spherical coordinates relative to the position of the A-th nucleus. The angular functions $Y_{lm}(\theta, \phi)$ are spherical harmonics. L denotes the maximum angular momentum number l of the expansion. Different choices for the actual form of the radial basis functions are possible[1–4].

The main reason for the success of the methods based on atomic orbitals is the compactness of the basis set; the exact wave function can be qualitatively approximated using basis sets containing a small number of functions. For instance, the ground state of H_2 can be modelled using just two s ($L = 0$) Slater-type orbitals (STO). At self-consistent-field (SCF) levels, the error in the energy with respect to the complete basis set decreases like e^{-L}, whereas a much slower decay like $(L+1)^{-3}$ is observed at *ab initio* correlated levels[5–7]. The accuracy of the results can be improved by extending the basis set, both in terms of increasing L and by adding more radial functions. However, the basis-set convergence is neither smooth nor fast, even though increasing L yields systematically smaller contributions. Enlarging the basis set with a large number of radial functions is not straightforward, because in practice it requires optimizations that might bias the basis set for some applications. Even though procedures to minimize this problem exist[8], an atom-centered AO basis set cannot be arbitrarily large, because of the onset of numerical instabilities caused by linear dependencies.

An alternative approach is to use local basis sets that consist of functions with compact support, such as finite-element functions. The local representation of the orbitals can be systematically improved in the regions where necessary without encountering problems with linear dependencies. The most precise electronic structure calculations have been performed in this way.[9–17] However, the molecular orbitals and the electronic interaction potentials are very steep in the vicinity of the nuclei due to the singularity of the nuclear potentials. A huge number of local basis functions must therefore be used to describe these sharp features.

Hence, local basis-set methods involve a large number of similar basis functions, which are spatially repeated. The type of operations involved are well suited for parallelization on SIMD (single instruction multiple data) architectures such as general-purpose graphics processing units (GPGPU), whereas global basis-set methods rely on diagonalization of very large matrices and integral algorithms that cannot easily be parallelized, especially on GPGPUs. Although remarkable progress has been made in this direction[18–20], the algorithms are not straightforward, and the performance for basis sets with large L is not clear. On the other hand, very efficient implementations of real-space electronic structure methods are becoming increasingly common on GPGPUs.[21–25]

Recently, we proposed a numerical scheme to represent the scalar functions encountered in electronic structure calculations, such as orbitals, electron densities and potentials.[26] The functions are decomposed into atom-centered parts (the *bubbles*), which contain the steep contributions, and a remainder represented in a Cartesian grid (the *cube*). The bubbles are written as products of angular parts, represented by real spherical harmonics, and radial parts, which are expanded in one-dimensional finite-element functions. A similar partition in terms of Gaussian functions and finite elements can be found in the literature.[27,28] We showed that it is possible to partition molecular charge densities such that the remainder is smooth and can be accurately represented with a three-dimensional grid of tractable size (ca. 200-400 points per dimension).[26] We also described algorithms to perform the two crucial operations required for performing SCF calculations, namely computing products of scalar functions and computing the action of operators that can be written in a Cartesian-separable form,[26]

$$g(\mathbf{r}) = \hat{O} f(\mathbf{r}) \approx \sum_p \omega_p \hat{O}^{x,p} \hat{O}^{y,p} \hat{O}^{z,p} f(\mathbf{r}). \tag{2}$$

The Laplacian operator, $\nabla^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$, and convolutions with the Helmholtz kernel,

$$g(\mathbf{r}) = \int_{\mathbb{R}^3} \frac{e^{-\kappa r'}}{|\mathbf{r} - \mathbf{r}'|} f(\mathbf{r}') \mathrm{d}^3 r', \tag{3}$$

are examples of operators that can be written in the form of Eq. (2). Parallelizing the two operations by means of SIMD architectures is completely straightforward.

In this work, we describe GPGPU-based implementations of the two operations, and assess their performance. The numerical representation of functions and the most expensive

computational operations are described in Section II. The implementation and benchmarks on GPGPUs, including test calculations on molecular systems, are discussed in Section III. We summarize the obtained results including a future outlook in Section IV.

The calculations have been performed on NVIDIA Tesla M2050 and M2070 graphics cards. Both cards have a 515 GFLOPs theoretical peak performance in double precision. The difference is the amount of memory available, 2.625 GB for the M2050 and 5.25 GB for the M2070. CPU calculations were carried out for comparison on an Intel Xeon X5650 (2.67 GHz), using a single core. However, we will not emphasize the difference in performance, as more effort was invested on optimizing the GPGPU version.

II. THE BUBBLES REPRESENTATION

A. Numerical representation of three-dimensional scalar functions

The bubbles representation can be summarized as follows. For a system consisting of K atoms, a function $f(\mathbf{r})$ is partitioned as

$$f(\mathbf{r}) = \sum_{A=1}^{K} f^A(r_A, \theta_A, \phi_A) + f^\Delta(\mathbf{r}) \tag{4}$$

where the atom-centered functions $f^A(r_A, \theta_A, \phi_A)$ are the *bubbles* and the remainder $f^\Delta(\mathbf{r})$ is the *cube*. The bubbles consist of radial and angular parts:

$$f^A(r_A, \theta_A, \phi_A) = \sum_{l=0}^{L} \sum_{m=-l}^{l} f^{Alm}(r_A) Y_{lm}(\theta_A, \phi_A) \tag{5}$$

L is typically small, e.g., 1 or 2 corresponding to p and d AOs. The angular part of the bubbles are real spherical harmonics, $Y_{lm}(\theta, \phi)$, which are stored in their Cartesian representation

$$Y_{lm}(\theta, \phi) = \sum_{uvw} C^{lm}_{uvw} \left(\frac{x}{r}\right)^u \left(\frac{y}{r}\right)^v \left(\frac{z}{r}\right)^w . \tag{6}$$

Expressions to compute the C^{lm}_{uvw} coefficients can be found in the literature.[29] For brevity, the spherical harmonics centered at point $\mathbf{R}_A$ are denoted as $Y^A_{lm} \equiv Y_{lm}(\theta_A, \phi_A)$.

Both the radial functions $f^{Alm}(r_A)$ and the cube $f^\Delta(\mathbf{r})$ are represented in terms of tensorial finite elements, that is

$$f^{Alm}(r_A) = \sum_i f_i^{Alm} \chi_i^A(r_A) \tag{7}$$

and

$$f^{\Delta}(\mathbf{r}) = \sum_{ijk} f_{ijk}^{\Delta} \chi_i^x(x) \chi_j^y(y) \chi_k^z(z) \tag{8}$$

In the notation introduced throughout Eqs. (4)-(8), for a function designated by a symbol f, the different pieces that constitute it are denoted by superscripts (f^A, f^{Δ}, f^{Alm}), and the expansion coefficients into which functions are expanded are denoted by subscripts (f_{ijk}^{Δ}, f_i^{Alm}).

The tensorial finite-element basis is described in the following. The one-dimensional finite-element basis is composed of a set of functions $\{\chi_i(x) | 1 \leq i \leq N\}$, each associated to a grid point x_i. The grid is constructed by subdividing the calculation domain $[x_{\min}, x_{\max}]$ into M intervals, or *cells*. Each cell is further subdivided into P subintervals of equal length. The length of the subintervals, or *step*, of the j-th cell is denoted by h_j. The grid points $\{x_i\}$ are located at the limits of these subintervals. The last grid point in one cell is also the first grid point of the next one, and the total number of points is $N = MP + 1$. The interval spanned by the j-th cell is $[s_j, s_j + Ph_j)$, where $s_j = x_{\min} + P \sum_{k=1}^{j-1} h_k$ is the first point in one cell, such that $s_j = x_{(j-1)P+1}$.

In each cell, a P-th order Lagrange interpolating polynomial (LIP) basis is constructed

$$b_{ij}(x) = \begin{cases} \displaystyle\prod_{\substack{0 \leq k \leq P \\ k \neq j}} \frac{(x - s_i)/h_i - k}{j - k} & \text{if } s_i \leq x < s_i + Ph_i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

with $1 \leq i \leq M$ and $0 \leq j \leq P$.

To ensure continuity, the basis is constructed so that functions located at junctions between cells spread over two adjacent cells:

$$
\begin{aligned}
\{\chi_i(x)|1 \leq i \leq MP+1\} = \{&\chi_1(x) = b_{10}(x), \\
&\chi_2(x) = b_{11}(x), \ldots, \\
&\chi_{P+1}(x) = b_{1P}(x) + b_{20}(x), \\
&\chi_{P+2}(x) = b_{21}(x), \ldots, \\
&\chi_{2P+1}(x) = b_{2P}(x) + b_{30}(x), \ldots, \\
&\chi_{MP+1}(x) = b_{MP}(x)\}
\end{aligned}
\tag{10}
$$

Since $\chi_i(x_j) = \delta_{ij}$, the expansion coefficient for $\chi_i(x)$ is the value of the function at the corresponding grid point x_i, i.e.

$$
f_i = f(x_i). \tag{11}
$$

A basis set in the range of $[-1.5, 1.5]$ with $M = 2$, $P = 2$ and steps $H = \{0.5, 1\}$ is illustrated in Fig. 1. In the rest of this work, LIPs of order $P = 6$ are used.

In the one-dimensional case, the function can be directly represented as a piece-wise sum of polynomials to speed up evaluation:

$$
f(x) = \sum_{i=0}^{M} w_i(x) \sum_{j=0}^{P} a_{ij} \left(\frac{x - s_i}{h_i} \right)^j \tag{12}
$$

where

$$
w_i(x) = \begin{cases} 1 & \text{if } s_i \leq x < s_i + Ph_i \\ 0 & \text{otherwise} \end{cases} \tag{13}
$$

The coefficients a_{ij} are such that $w_i(x) \sum_{j=0}^{P} a_{ij}((x - s_i)/h_i)^j = \sum_{j=0}^{P} f_{(i-1)P+j} b_{ij}(x)$. However, the f_i expansion coefficients are stored, which simplifies the calculation of the expansion coefficients of new functions by using inner projection, such as $g(f(x)) = \sum_i g(f_i)\chi_i(x)$ or $g(x)f(x) = \sum_i f_i g_i \chi_i(x)$.

For a higher number of dimensions, the grid is generated as a Cartesian product of one-dimensional grids. For instance, given $X = \{x_i|1 \leq i \leq N_x\}$, $Y = \{y_j|1 \leq j \leq N_y\}$ and $Z = \{z_k|1 \leq k \leq N_z\}$, the three-dimensional grid is constructed as

$$
\{\mathbf{r}_{ijk} = (x_i, y_j, z_k)\} = X \times Y \times Z. \tag{14}
$$

The basis set is then a tensor product of the one-dimensional bases $\{\chi_i^x(x)|1 \leq i \leq N_x\}$, $\{\chi_j^y(y)|1 \leq j \leq N_y\}$ and $\{\chi_k^z(z)|1 \leq k \leq N_z\}$:

$$\begin{aligned} &\{\chi_{ijk}(x,y,z) = \chi_i^x(x)\chi_j^y(y)\chi_k^z(z)\} = \\ &\{\chi_i^x(x)\} \otimes \{\chi_j^y(y)\} \otimes \{\chi_k^z(z)\} \end{aligned} \tag{15}$$

The basis set of the radial functions (see Eq. (7)) is constructed on a non-equidistant grid, which provides a sufficiently good representation so that it will not pose a bottleneck in accuracy. The cut-off radius $r_{\max} = 20$ a$_0$ and the number of cells is $M_A = 200Z_A^{1/4}$, Z_A being the nuclear charge of the A-th atom. The details on how the grid is generated have been discussed elsewhere.[26] In practical calculations, it is desirable to set the cut-off radii as small as possible, to minimize the amount of overlap between bubbles centered at different atoms. A certain degree of overlap is nevertheless required. Otherwise the remainder to be represented in the cube is too large in the interstitial regions, which worsens the accuracy.

The radial functions can be extended to include some analytical factor, i.e.

$$f^{Alm}(r_A) = g(r)\sum_{i=1}^{M} f_i^{Alm}\chi_i(r_A). \tag{16}$$

For instance, $g(r) = r^{-1}$ or $g(r) = r^{-2}$ can be used to explicitly include singularities at the origin of the bubbles when representing functions like nuclear potentials or Laplacians, akin to atomic structure calculations.[9,30]

The cubes are represented in a single three-dimensional tensorial basis. Using a single Cartesian grid to represent the whole cube has a drawback: unnecessary points must be unavoidably stored, as illustrated in Fig. 2. This can be mitigated by partitioning the domain into smaller grids. However, this type of partitioned basis is not tensorial anymore, which eliminates the separability into one-dimensional functions. Using one single tensorial grid permits a very efficient parallel implementation of certain crucial operations, as will be discussed below.

The cube grid ranges are chosen such that the domain boundaries are at least 8 a_0 apart from any atomic center. In this work, we use equidistant grids. Exploring more efficient non-equidistant grids is beyond the scope of this work although, based on some preliminary calculations, we estimate that the number of grid points per dimension can be cut down by a factor two, without affecting the accuracy.

The memory requirements are largely determined by the size of the cube. For a typical grid step of 0.1 a_0, the total number of cube grid points per dimension is 100-200, for a total of $N_{\text{cub}} = N_x N_y N_z$ of $10^6 - 10^7$ coefficients (10-100 MB using 64-bit floating-point numbers). The storage requirements of the bubbles are very small compared to the total memory used, amounting to $N_{\text{bub}} = \sum_A (L+1)^2 (M_A P + 1)$ coefficients. For $L = 2$, the number of coefficients per atom ranges between $5 \cdot 10^3$ and $1.5 \cdot 10^4$ corresponding to about 100 kB per atomic center. As the volume of the cube grows approximately linearly with the number of atoms as $aK + b$, so does the storage cost, as illustrated in Fig. 3 for grids with a step of 0.1 a_0, for some molecular test systems (see Sec. III C). An "empty box" of roughly $16 \times 16 \times 16$ a_0^3 would require $b = 4.40 \cdot 10^6$ coefficients (ca. 40 MB for 64 bit floats), and each atom added requires $a = 6.25 \cdot 10^5$ coefficients more (approximately 5 MB per atom). Halving the grid step increases the number of coefficients approximately by a factor of 8.

B. Two-function products

The first computational bottleneck arises when computing products of two functions. Products happen often, as for instance in the computation of charge densities $\rho = \sum_i |\phi_i|^2$, dot products such as $\langle f, g \rangle$, or when multiplying a potential with an orbital. Although functions represented in the bubbles basis can be accurately multiplied in a point-wise manner, a more desirable approach is to obtain the resulting function directly in the bubbles representation.

Let us consider consider the product $f(\mathbf{r})g(\mathbf{r}) = h(\mathbf{r})$, which can be expanded as

$$\Big[\sum_A f^A(r_A, \theta_A, \phi_A) + f^\Delta(\mathbf{r})\Big]\Big[\sum_A g^A(r_A, \theta_A, \phi_A) + g^\Delta(\mathbf{r})\Big] = \sum_A h^A(r_A, \theta_A, \phi_A) + h^\Delta(\mathbf{r}). \quad (17)$$

We recently proposed a scheme to directly compute the radial parts of the resulting function such that the resulting cube $h^\Delta(\mathbf{r})$ is sufficiently smooth.[26] The radial functions are obtained as:

$$h^{Alm}(r_A) = \sum_{l_1 m_1} \sum_{l_2 m_2} \langle Y_{lm} | Y_{l_1 m_1} Y_{l_2 m_2} \rangle \Big[f^{Al_1 m_1}(r_A) \tilde{g}^{Al_2 m_2}(r_A) \mu(r_A) + \tilde{f}^{Al_1 m_1}(r_A) g^{Al_2 m_2}(r_A) \nu(r_A) \Big]. \quad (18)$$

The radial functions $\tilde{f}^{Alm}(r_A)$ in Eq. (18) are obtained by expanding $f - f^A = \sum_{B \neq A} f^B + f^\Delta$ around $\mathbf{R}_A$ in a truncated Taylor series of order T, which is projected onto the spherical harmonics basis. The same procedure is used to compute $\tilde{g}^{Alm}(r_A)$. The $\mu(r)$ and $\nu(r)$ functions in Eq. (18) enforce a correct long-range behaviour and depend on the properties of $f(\mathbf{r})$ and $g(\mathbf{r})$. Here, we consider the case $\mu(r) = 1$ and $\nu(r) = 0$, valid for computing products of densities and non-singular potentials.

The L value of the product function is the sum of the L values of the multiplicands (or the order of their Taylor series, whichever is larger), although it can be truncated. In such case, all the discarded higher l functions would then be represented in the cube.

Obtaining the radial parts as in Eq. (18) is computationally simple. The coefficients of the product function in the cube are then computed point-by-point,

$$h^\Delta_{ijk} = \left[f^\Delta_{ijk} + \sum_{A=1}^{K} f^A(x_i, y_j, z_k) \right] \left[g^\Delta_{ijk} + \sum_{A=1}^{K} g^A(x_i, y_j, z_k) \right] - \sum_{A=1}^{K} h^A(x_i, y_j, z_k). \qquad (19)$$

This is the main computational bottleneck, because it requires evaluating the bubbles of each of $f(\mathbf{r})$, $g(\mathbf{r})$ and $h(\mathbf{r})$ at every grid point of the cube. We denote this operation as *bubbles injection*.

The algorithm for the bubbles injection is outlined in Fig. 4. First, we precompute the coefficients of the interpolating polynomials as of Eq. (12). The computational cost for this is negligible. Then, for every grid point, the bubble is interpolated. This consists of a series of steps, namely computing the relative coordinates of the grid point respect to the bubble center, find the cell in the radial grid, transform to cell coordinates, and, for every l and m, the radial function is interpolated and the spherical harmonic evaluated.

Due to the large number of cube grid points, this operation is very costly. Moreover, it must be done three times, for each one of the $f(\mathbf{r})$, $g(\mathbf{r})$ and $h(\mathbf{r})$ functions. However, all loops can be run in any order. In particular, the loop over cube grid points requires no communication whatsoever, making the operation embarrassingly parallel. Moreover, the operations for different grid points are exactly identical, making them ideal for execution on SIMD architectures. This is also true for finding the cell in the radial grid, as it can be done without conditionals by performing a binary search with $\log_2(M_A + 1)$ comparisons. The total cost for the injection is in practice proportional to the number of interpolations, $N_{\text{cub}} K (L+1)^2$. Owing to the linear growth of N_{cub} with K, as was shown in Fig. 3, the

computational cost is expected to be proportional to $(aK+b)K(L+1)^2$. Although formally $\mathcal{O}(K^2)$, the quadratic behavior will only be noticeable for large systems, owing to b being one order of magnitude larger than a. Furthermore, linear scaling can be achieved by means of smaller cut-offs for the radial grids and neighbor lists.

It must be remarked that the computational costs hardly increase with the number of radial grid points, implying that more accurate radial grids can be used when needed.

C. Linear transformations

Linear transformations appear in several circumstances, such as computing the Laplacian of a function or convoluting with the Poisson kernel, i.e.

$$V(\mathbf{r}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \mathrm{d}^3 r' \tag{20}$$

The effect of linear operators can be computed independently for the bubbles and the cube

$$\hat{O}f(\mathbf{r}) = \sum_{A=1}^{K} \hat{O}f^A(r_A, \theta_A, \phi_A) + \hat{O}f^\Delta(\mathbf{r}). \tag{21}$$

Linear operators can often be expressed in spherical coordinates, implying that their effect on the bubbles can be computed very efficiently and accurately. For instance, for the bubbles, the radial parts resulting from a convolution with the Poisson kernel are given by[30]

$$V^{Alm}(r_A) = \frac{4\pi}{2l+1}\left[{r_A}^{-(l+1)} \int_0^{r_A} \rho^{Alm}(r'_A) {r'_A}^{l+2} \mathrm{d}r'_A \; + {r_A}^{l} \int_{r_A}^{\infty} \rho^{Alm}(r'_A) {r'_A}^{1-l} \mathrm{d}r'_A \right]. \tag{22}$$

The cost for computing the integrals in Eq. (22) in the finite-element basis is negligible.

The effect of the operator on the cube functions can be computed efficiently when the operator can be approximated in a Cartesian-separated form of rank R:

$$\hat{O} \approx \sum_{p=1}^{R} \omega_p \hat{O}^{x,p} \hat{O}^{y,p} \hat{O}^{z,p} + C\hat{I} \tag{23}$$

where R is the *rank* of the operator, $\hat{I}$ is the identity operator, and C and $\{\omega_p | 1 \leq p \leq R\}$ are constants. For such a transformation, the coefficients of g^Δ can then be obtained as

$$g^{\Delta}_{ijk} \approx \sum_{p=1}^{R} \omega_p \sum_{k'=1}^{N_z} O^{z,p}_{kk'} \sum_{j'=1}^{N_y} O^{y,p}_{jj'} \sum_{i'=1}^{N_x} O^{x,p}_{ii'} f^{\Delta}_{i'j'k'} + C f^{\Delta}_{ijk} \tag{24}$$

The elements of the $\mathbf{O}^p$ matrices are

$$O^{\xi,p}_{ii'} = \hat{O}^{\xi,p} \chi_{i'}(\xi)\Big|_{\xi=\xi_i} \tag{25}$$

It should be noted that the operators can have different input and output grids, the i', j' and k' indices referring to elements of the input grid and the i, j and k indices to elements of the output grid.

The rank R, the explicit form of the matrix elements of the operator matrices, the ω_p values, and C coefficient depend on the operator in question. For example, for the Laplacian (∇^2), Eq. (24) holds exactly, with $R = 3$ and $C = 0$. For convolutions with the Poisson kernel, we have previously shown the operator can be approximated with a rank of $R = 20$ to obtain potentials of the studied systems with 6-8 digit accuracy.[26,31] In our scheme, the Coulomb potential is approximated using a numerical quadrature[26] of the well-known integral identity[32,33]

$$\frac{1}{r_{12}} = \frac{2}{\sqrt{\pi}} \int_0^{\infty} e^{-t^2 r_{12}^2} \mathrm{d}t \approx \sum_{p=1}^{R} \omega_p e^{-t_p^2 r_{12}^2} + \frac{\pi}{t_f^2} \delta(\mathbf{r}_1 - \mathbf{r}_2), \tag{26}$$

The details on how to obtain the points t_p and weights ω_p can be found in our previous work.[26,31]

The expression in Eq. (24) can be written as a series of matrix multiplications. Following the notation of Kolda et al.,[34] we denote the three-index tensor containing the elements f^{Δ}_{ijk} as $\mathbf{F}$, and the tensor of the cube of the output function as $\mathbf{V}$. $\mathbf{G}$ and $\mathbf{H}$ are respectively intermediate two- and three-dimensional tensors. Two-index slices are expressed as e.g. $\mathbf{F}_{:i:}$, which is a matrix whose elements are $(\mathbf{F}_{:i:})_{jk} = (\mathbf{F})_{jik}$. The operator matrices are stored as three-dimensional tensors $\mathbf{O}^x$, $\mathbf{O}^y$ and $\mathbf{O}^z$, with $(\mathbf{O}^{\xi})_{i,i',p} = O^{\xi,p}_{ii'}$. The algorithm is outlined in Fig. 5. The operation consists of a series of matrix multiplications. The matrix multiplications are carried out in the order x, y then z, because the elements of the cube and the operator matrices are stored in column-major order. The total number of floating-point operations for a cubic grid with $N = N_x = N_y = N_z$ is approximately $N^3(6RN+2) \sim 6RN^4$. Because of the linear increase of N_{cub} with the number of atoms K, the computational cost is expected to grow proportionally to $R(aK + b)^{4/3}$.

III. GPGPU IMPLEMENTATION AND BENCHMARKING

In this section, we describe the GPGPU implementation of the operations described in Sections II B and II C, and we assess their performance using some model systems. We also present timings for the same molecular systems computed in our previous work.[26] All timings include the transfer of data to the GPU and back to the main memory.

A. Products

As discussed in Sec. II B, the injection operation is embarrassingly parallel over cube grid points. Its implementation as a CUDA C kernel is straightforward, involving no changes to the algorithm, besides using the minimum number of registers possible. Initially, the required data structures (spherical harmonics, cube grid) are transferred to the GPU, and the cube is allocated. Then, for each bubble, the radial grids and expansion coefficients are copied to the GPU, and the injection is carried out for that bubble. Finally, the resulting cube is copied back to main memory. Having all threads in the GPU injecting the same bubble minimizes the required amount of global memory. If the hardware allows it (e.g., multiple GPUs are available), the injection can be launched simultaneously for several bubbles.

The parallelization is done by subdividing the cube grid into blocks. Blocks of $8 \times 4 \times 4$ points provides maximal occupancy, i.e., yields the largest possible number of simultaneously running threads on the M2050/M2070 GPUs. Each block is evaluated by a CUDA block of 128 threads, each thread taking care of one point. The main performance limitation is the lack of memory coalescence. In other words, each thread most probably needs a different interpolation cell and has to fetch a different chunk of the radial function from global memory. We considered the possibility of using shared memory to overcome this issue. However, due to the size of the data structures, this approach severely limited the number of simultaneous running threads, which lowered the overall efficiency.

The performance of the injection was tested for different K, L and $N = N_x = N_y = N_z$ values with $Z = 1$. As a measurement of the performance, we use the number of interpolations per second, computed as $N^3K(L+1)^2$ divided by the total computing time. Note that an actual two-function product requires three injections. The performance is shown in Fig. 6 as a function of the number of cube grid points per dimension N. The performance

is saturated for grids of size about $150 \times 150 \times 150$, $K = 5$ and $L = 2$ with a rate of 1.5 to 1.7 billions of interpolations per second. The total computational time is approximately $N^3K[0.8(L+1)^2 + 2.4] \cdot 10^{-9}$ s. E.g., for a $200 \times 200 \times 200$ cube and using $L = 2$ the cost is expected to be about 0.08 s per atom.

Because of the presence of comparisons, square roots and exponentiations, the FLOP count for this operation is difficult to establish. Hence, estimating the performance in GFLOPs or as a percent of the theoretical peak performance of the GPGPU is difficult for this operation. Moreover, the number of interpolations per second can be unambiguously measured for any kind of architecture, which is not the case for FLOP counts. In any case, memory access seems to be the limiting factor, and therefore GFLOPs is most likely a poor measurement of the performance of this step. A rough estimate can nevertheless be obtained by assuming that 0.8 ns is needed for evaluating one polynomial and one spherical harmonic, which corresponds to ca. 20 FLOP when considering exponentiation as one FLOP. The estimated computational speed of 25 GFLOPs is about 5% of the theoretical peak performance of the GPGPU.

Compared to the single CPU core version, the GPGPU implementation is roughly two orders of magnitude faster. This comparison should though be taken with a pinch of salt, as the CPU version is a straightforward non-optimized implementation, as discussed earlier. However, this clearly shows how suitable the algorithm is for SIMD architectures: a relatively trivial implementation, involving little optimization, is much faster than the serial version can possibly be.

B. Linear transformations

We based our GPGPU implementation of the linear transformations on the cuBLAS library, which is NVIDIA's GPGPU port of the widespread BLAS library[35]. Matrix algebra operations are typically repetitive, and therefore amenable for implementation on GPGPUs. Indeed, GPGPU-accelerated matrix multiplications can be found in the context of quantum chemical calculations.[36–39] In our implementation, we first transfer the input tensors ($\mathbf{F}$, $\mathbf{O}^x$, $\mathbf{O}^y$ and $\mathbf{O}^x$), and allocate the output tensor $\mathbf{V}$ and the intermediate tensors $\mathbf{G}$ and $\mathbf{H}$. The total memory requirements are $3N^3 + (3R+1)N^2$ coefficients. Then, the R series of multiplications are carried out, and $C\mathbf{F}$ is added. Finally, the output tensor is downloaded

to main memory. We tested the cuBLAS threaded version of the matrix multiplication, parallelizing over R. However, this did not significantly affect performance. This is possibly due to the need to allocate larger intermediate tensors, which limits the amount of available memory and hence the number of concurrent threads. Parallelization over multiple GPGPUs is expected to scale properly, and will be considered in future implementations. No other parameters controlling the execution were modified.

We benchmarked the linear transformation for cubic grids of different N up to 361 ($M = 60$), and R ranging from 1 to 60. The performance in GFLOPs is shown in Fig. 7 for some selected values of R. Performance increases both with R and N, in other words, as the calculation becomes heavier and the ratio of computing time to data transfer increases. At approximately $N = 150$, there is a significant jump in the efficiency. This is due to cuBLAS switching to a different algorithm. The performance is practically saturated for $N = 200$ for $N \geq 200$ and $R \geq 20$, at ca. 250-300 GFLOPs (50-60% of the M2070 peak performance). For these cases, the computational times can be approximated as $1.2 \cdot 10^{-8} N^3 + 2.1 \cdot 10^{-11} R N^4$ s. For a cube of $200 \times 200 \times 200$ and $R = 20$, this is about 0.8 s.

We measured a speed-up of about 30-50 relative to the single CPU core version, which was implemented using the BLAS library. This compares well with the speed-ups announced by the vendor (a factor of 5 compared to a hex-core processor with a similar clock speed as the one used here). It seems clear that the present algorithm is very suitable for GPGPUs. This further supports the idoneity of the unique Cartesian grid for GPGPU implementations. A very fine domain-partitioning, i.e., one cell per process, would have resulted into a very large number of multiplications of very small $(P + 1) \times (P + 1)$ matrices, which, according to Fig. 7, would be much less efficient.

It should be remarked that typically one would not expect so good performance on GPGPUs for matrices of the size discussed here. For instance, in the work of Olivares-Amaya et al.[37] it is clear that, in order to fully use the power of the GPGPU, matrices of several thousand elements per dimension need to be used. The reason for the excellent performance obtained here is that there is a very large amount of matrix multiplications ($3RN$) computed in series, which supposes a sufficiently large computational workload between GPU-CPU transfers.

V

C. Calculations on test molecular systems

In our previous work, we assessed the accuracy of the present computational approach using charge densities obtained in calculations on a set of 19 molecules, including up to 15 atoms.[26] The electron densities were computed with the ADF program suite[40–42] in the local density approximation (LDA) using a double-ζ STO basis set and a large frozen core. The charge densities are partitioned using an ad-hoc algorithm which is expected to yield sufficiently smooth remainders.[26]

The target quantity chosen to assess the accuracy of the representation is the total Coulomb energy, consisting of the electron-electron plus the electron-nuclear interactions. Starting from the electron density partitioned into bubbles and cubes, the required steps are one convolution with the Poisson kernel (see Sections II B and III A) and the product of the resulting electrostatic potential with the electron density (see Sections II C and III B), and finally the integration of the resulting energy density. The nuclear-electronic interaction is straightforwardly computed from the electrostatic potential as

$$E_{\text{nuc}} = -\sum_A Z_A V(\mathbf{R}_A) \tag{27}$$

The results of the calculations on GPGPU are summarized in Table I. For each molecule we report the results for grids with steps of 0.4, 0.2, 0.1 and 0.05 a_0, in descending order.

Columns N_{bub} and N_{cub} contain the amount of coefficients required to store the bubbles and the cubes, respectively. As was shown in Fig. 3 for the 0.1 a_0 grid, N_{cub} grows linearly with the number of atoms. When the grid step is halved, N_{cub} roughly increases by a factor of 8. N_{bub} grows linearly with K, although in all cases it is much smaller that N_{cub}.

In column ΔE we report the difference with respect to the total Coulomb energy, as reported by ADF with a 6-digit integration accuracy. Except for the smallest systems, the two coarsest grids (with steps of 0.4 and 0.2 a_0) are not sufficient to obtain sub-mE_{h} accuracy. For more than half of the systems, this accuracy is obtained with the 0.1 a_0 step grid, while the 0.05 a_0 step grid is needed to push all the errors below the 1 mE_{h} threshold (with the exception of C_6F_6, $\Delta E = 1.5$ mE_{h}). The error does not decay as fast for different molecules, yielding errors all over the $\mu E_{\text{h}} - \text{m}E_{\text{h}}$ range (note that the reference values were computed with 6-digit precision). This is due to the irregular performance of the employed ad-hoc partitioning scheme.

The last two columns contain the cost of the benchmarked operations in seconds. In parenthesis, we report the cost per "elemental operation", which we define as one interpolation for the injection and one floating-point operation for the convolution. The time required to convolute the density with the Coulomb kernel with $R = 20$ is given in column t_*, with the cost per operation calculated as $t_*/(120N_{\text{cub}}^{4/3}) \cdot 10^{-12}$ (roughly seconds per 10^{12} floating-point operations). In column $t_\times$ we report the cost of the product of the density with the potential, with the time per operation calculated as $t_\times/(17KN_{\text{cub}}) \cdot 10^{-9}$ (seconds per billion of interpolations). The multiplication involves the injection of three functions, two of them with with $L = 1$ and one function with $L = 2$ (hence the factor of $17 = 2(1+1)^2 + (2+1)^2$, two cube additions, one cube product and one cube subtraction (see Eq. (17)). In the estimate of the cost per operation for the product we have only considered the bubble injections, although the cube arithmetics account for roughly 10% of the total cost of the product.

For a given grid step, the computational costs for different molecules are comparable, although $t_\times$ grows faster than t_* with respect to the number of atoms ($(aK + b)K$ vs. $(aK + b)^{4/3}$). Nevertheless, the costs of both operations are quite similar, differing at most by a factor of about 4.

The cost per elemental operation, i.e., the performance, is almost constant for different molecules, when the same grid is used. This indicates that the computing time can be extrapolated to other systems quite accurately. The performance of both operations improves as a finer grid is used, as was observed previously in the benchmarks, due to the increase in the ratio of computational workload to data transfer. The improvement is particularly clear for the convolution, which performs quite poorly for the smaller grids. The maximum performance obtained (for adenine with the 0.05 a_0 grid) corresponds to 294 GFLOPs for the convolution and $1.25 \cdot 10^9$ interpolations per second for the multiplication. This correlates well with the peak performances observed in Figures 6 and 7.

We expect that the combination of an improved partitioning scheme together with more judicious non-equidistant grids should bring the accuracy of the 0.05 a_0 grids (fourth rows of each entry) at the cost of the 0.1 a_0 grids (third rows). Such set up would be optimal, as the 0.1 a_0 grids, which consist of ca. 180^3-240^3 points, already show almost maximal performance.

V

IV. CONCLUSIONS AND OUTLOOK

In this paper, we have presented and benchmarked GPGPU implementations of the costliest parts of our fully numerical electronic structure framework *bubbles*. The present work offers evidence for the suitability of numerical methods for the emerging parallel computer architectures. The two bottlenecks addressed are the multiplication of two three-dimensional scalar functions, and the application of certain linear transformations, such as the convolution with the Poisson kernel. Due to the large number of cube grid points, these operations are computationally very time-consuming. However, owing to the Cartesian separability of the basis, the algorithms are trivially parallelizable and very suitable for GPGPU architectures. The GPGPU implementation was straightforward, either by means of available libraries, or by developing our own CUDA kernels. There was a very significant decrease (a factor of 30-100) in the cost of the operations with respect to the serial CPU implementation. The benchmarking showed very good performance when the grids were sufficiently large. Grids smaller than approximately 180^3 points should not be used, as they are not able to saturate the GPU bandwidth.

The available GPGPU hardware seems to be already ideal for the algorithms presented, both in terms of computational power and available memory, as the implementation required no modification of the original CPU algorithms. Of course, improvements in cache, bandwidth, memory, clock speed, etc. in newer generations of graphics cards will do nothing but increase the performance of the presented methods. The amount of available on-card memory should not pose a limitation. For all systems considered, the operations can be carried out in the 5.25 GB of memory of the M2070 card. The largest system presented here is adenine on a grid with a step of 0.05 a_0, which consists of $541 \times 505 \times 385$ grid points and occupies about 801 MB. This calculation is close to the limit of about 500^3 grid points (ca. 1 GB) that can be handled on a single GPGPU. This limit is imposed by the linear transformation, which requires the allocation of three cubes of that size, plus additional overheads required by cuBLAS. Ultimately, due to the embarrassingly parallel character of the algorithms, the operations could be computed in several stages if needed or, even better, on multiple GPGPUs running simultaneously. Memory is not a limitation: if the number of cube grid points per dimension can be cut by half, we can extrapolate from Fig. 3 that 200 atoms fit in one GPU. In any case, for systems of such size the computing time would be

excessive.

The most costly bottlenecks for the implementation of a Hartree-Fock SCF cycle have therefore been levelled. An SCF cycle based on the approach of Yanai et al.[16] comprises, for n occupied orbitals, $13n^2 + 6n$ multiplications and $n^2 + 2n + 2$ convolutions.[43] For the 0.1 a_0 grids, we can estimate that the cost of an SCF cycle on the same hardware would be 4 minutes for water ($n = 5$), 200 minutes for sulphuric acid ($n = 25$) and 500 minutes for adenine ($n = 35$). Despite the high cost compared to LCAO approaches, it should be kept in mind that this results are expected to be close to the complete basis set limit. The pieces missing to implement such a scheme are an accurate representation of products of molecular orbitals and nuclear potentials, and convolutions of the Helmholtz kernel with these functions. An alternative approach would be directly constructing the Fock matrix, as in our representation the kinetic energy integrals can be accurately computed. This would be more similar to the common LCAO methods, circumventing orthonormalization procedures and converging quickly to the best linear combination of starting orbitals. A combination of both methods as alternating steps could possibly provide an efficient and accurate all-electron numerical SCF implementation.

ACKNOWLEDGEMENTS

This research has been supported by the Academy of Finland through project (137460) and its Computational Science Research Programme (LASTU/258258). CSC – the Finnish IT Center for Science – is acknowledged for computer time, and the Magnus Ehrnrooth Foundation for travel money. The authors would also like to thank Susi Lehtola and Ville Havu for their help polishing the manuscript, Javier Gómez and Stig-Rune Jensen for the helpful discussions, and Elias Toivanen, Dou Du and Jonas Jusélius for support with the development and maintenance of the code. A. A.-G. acknowledges support from the National Science Foundation under award number OIA-1125087 as well as support from the Corning Foundation.

REFERENCES

[1]J. C. Slater, Phys. Rev. **36**, 57 (1930).

[2]S. F. Boys, Proc. Roy. Soc. London **200**, 542 (1950).

[3]V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Comput. Phys. Commun. **180**, 2175 (2009).

[4]B. Delley, J. Chem. Phys. **92**, 508 (1990).

[5]W. Klopper and W. Kutzelnigg, J. Mol. Struct. (Theochem) **135**, 339 (1986).

[6]W. Kutzelnigg and J. D. Morgan, J. Chem. Phys. **96**, 4484 (1992).

[7]D. P. Carroll, H. J. Silverstone, and R. M. Metzger, J. Chem. Phys. **71**, 4142 (1979).

[8]P. Manninen and J. Vaara, J. Comput. Chem. **27**, 434 (2006).

[9]D. Sundholm and J. Olsen, J. Chem. Phys. **94**, 5051 (1991).

[10]C. Froese Fischer, Comput. Phys. Commun. **4**, 107 (1972).

[11]E. A. McCullough Jr., Chem. Phys. Letters **24**, 55 (1974).

[12]E. A. McCullough Jr., J. Chem. Phys. **62**, 3991 (1975).

[13]L. Laaksonen, P. Pyykkö, and D. Sundholm, Int. J. Quant. Chem. **23**, 309 (1983).

[14]L. Laaksonen, P. Pyykkö, and D. Sundholm, Int. J. Quant. Chem. **23**, 319 (1983).

[15]J. Kobus, L. Laaksonen, and D. Sundholm, Comput. Phys. Commun. **98**, 346 (1996).

[16]T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, J. Chem. Phys. **121**, 2866 (2004).

[17]F. A. Bischoff and E. F. Valeev, J. Chem. Phys. **134**, 104104 (2011).

[18]I. S. Ufimtsev and T. J. Martinez, J. Chem. Theory Comput. **4**, 222 (2008).

[19]J. Kussmann and C. Ochsenfeld, J. Chem. Phys. **138**, 134114 (2013).

[20]A. V. Titov, I. S. Ufimtsev, N. Luehr, and T. J. Martinez, J. Chem. Theory Comput. **9**, 213 (2013).

[21]L. Genovese, M. Ospici, T. Deutsch, J.-F. Méhaut, A. Neelov, and S. Goedecker, J. Chem. Phys. **131**, 034103 (2009).

[22]H. Tomono, M. Aoki, T. Iitaka, and K. Tsumuraya, J. Phys.: Conf. Ser. **215**, 12121 (2010).

[23]X. Andrade, J. Alberdi-Rodriguez, D. A. Strubbe, M. J. T. Oliveira, F. Nogueira, A. Castro, J. Muguerza, A. Arruabarrena, S. G. Louie, A. Aspuru-Guzik, A. Rubio, and M. A. L. Marques, J. Phys.: Condens. Matter **24**, 233202 (2012).

[24]J. Yan, L. Li, and C. O'Grady, Comput. Phys. Commun. (2013), http://dx.doi.org/10.1016/j.cpc.2013.07.014.

[25]X. Andrade and A. Aspuru-Guzik, J. Chem. Theory Comput. **0**, null.

[26]S. A. Losilla and D. Sundholm, J. Chem. Phys. **136**, 214104 (2012).

[27]M. A. Watson and K. Hirao, J. Chem. Phys. **129**, 184107 (2008).

[28]M. A. Watson, Y. Kurashige, T. Nakajima, and K. Hirao, J. Chem. Phys. **128**, 054105 (2008).

[29]T. Helgaker, P. Jørgersen, and J. Olsen, *Molecular Electronic-Structure Theory* (John Wiley & Sons, Chichester, 2000).

[30]C. Froese Fischer, *The Hartree-Fock Method for Atoms: A Numerical Approach* (John Wiley and Sons, New York, 1977).

[31]S. A. Losilla, D. Sundholm, and J. Jusélius, J. Chem. Phys. **132**, 024102 (2010).

[32]K. Singer, Proc. R. Soc. A **258**, 412 (1960).

[33]S. F. Boys, Proc. R. Soc. A **258**, 402 (1960).

[34]T. G. Kolda and B. W. Bader, SIAM Review **51**, 455 (2009).

[35]L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley, ACM Trans. Math. Softw. **28**, 135 (2002).

[36]L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, and A. Aspuru-Guzik, J. Phys. Chem. A **112**, 2049 (2008).

[37]R. Olivares-Amaya, M. A. Watson, R. G. Edgar, L. Vogt, Y. Shao, and A. Aspuru-Guzik, J. Chem. Theory Comput. **6**, 135 (2010).

[38]M. Watson, R. Olivares-Amaya, R. G. Edgar, and A. Aspuru-Guzik, Comput. Sci. Eng. **12**, 40 (2010).

[39]a. E. DePrince and J. R. Hammond, J. Chem. Theory Comput. **7**, 1287 (2011).

[40]G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler, J. Comp. Chem. **22**, 931 (2001).

[41]C. Fonseca Guerra, J. G. Snijders, G. te Velde, and E. J. Baerends, Theor. Chem. Acc. **99**, 391 (1998).

[42]"ADF2010," SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands (2010).

[43]Stig Rune Jensen, private communication.

FIGURES

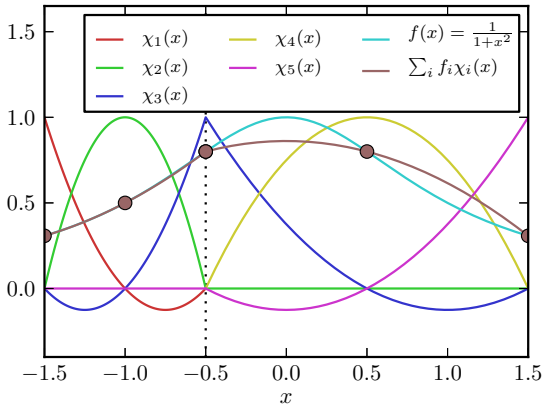


FIG. 1: The finite-element basis for $M = 2$ cells in the intervals $[-1.5, -0.5]$ and $[-0.5, 1.5]$ using a LIP order of $P = 2$. The cell steps are $H = \{0.5, 1\}$. The total number of basis functions is $N = 5$.

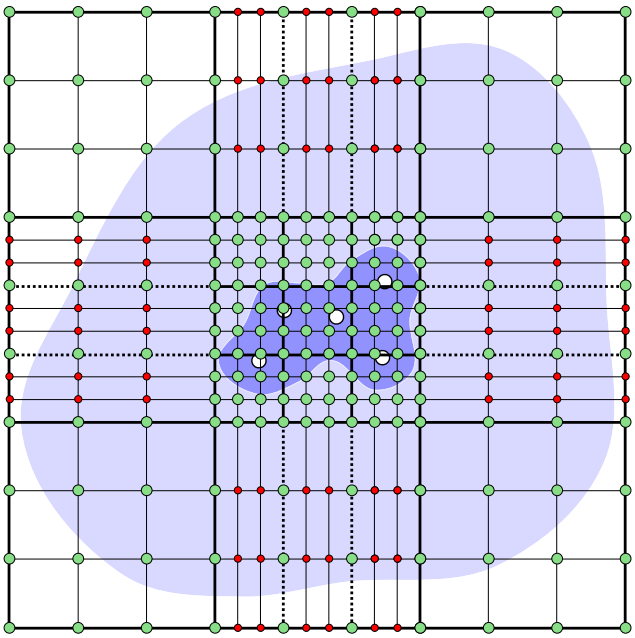


FIG. 2: A two-dimensional cross-section of a non-equidistant grid with a fine inner region and a coarse outer region. The red points are in practice superfluous, because these values must be stored, although they do not improve the overall accuracy of the calculation.

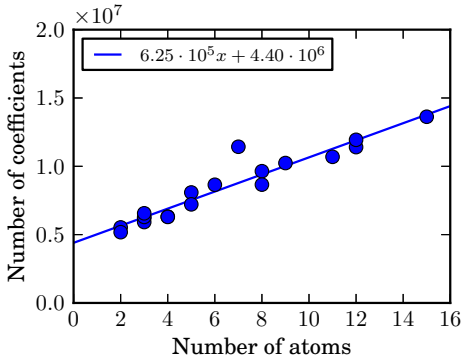


FIG. 3: Number of coefficients required to store the electron density as a function of the number of atoms, for the equidistant grids with a step of 0.1 a_0. The outlier is H_2SO_4. The data used for the plot is given in Table I.

for all A bubbles **do**

- *Precompute interpolating polynomials*

for all $l = 0, \ldots, L$ **do**

for all $m = -l, \ldots, l$ **do**

for all $i = 1, \ldots, M_A$ **do**

Compute a_{ij}^{Alm} such that

$$w_i(x) \sum_j a_{ij}^{Alm} ((x - s_i)/h_i)^j = \sum_j f_i^{Alm} b_{ij}(x)$$

- *Interpolate the bubbles*

for all $\mathbf{r}_{\alpha\beta\gamma} = (x_\alpha, y_\beta, z_\gamma)$ grid points **do**

- *Compute relative coordinates, distance, unit vector*

$\mathbf{r}_A \leftarrow \mathbf{r}_{\alpha\beta\gamma} - \mathbf{R}_A$

$r_A \leftarrow |\mathbf{r}_A|$

$\hat{\mathbf{r}}_A \leftarrow \mathbf{r}_A / r_A$

- *Find cell*

Find i such that $s_i \leq r_A \leq s_i + Ph_i$

- *Compute local coordinate*

$q \leftarrow (r_A - s_i)/h_i$

for all $1 \leq l \leq L$, $-l \leq m \leq l$ **do**

- *Evaluate interpolating polynomial*

$f \leftarrow \sum_{0 \leq j \leq P} a_{ij}^{Alm} q^j$

- *Evaluate spherical harmonic*

for all (u, v, w) such that $C_{uvw}^{lm} \neq 0$ **do**

$Y \leftarrow Y + C_{uvw}^{lm} \hat{x}_A^u \hat{y}_A^v \hat{z}_A^w$

- *Accumulate result*

$f_{\alpha\beta\gamma} \leftarrow f_{\alpha\beta\gamma} + fY$

FIG. 4: Algorithm for the bubbles injection for a function $f(\mathbf{r})$.

for all $1 \leq p \leq R$ **do**

  for all k' slices along the z axis **do**

    $\mathbf{G} \leftarrow \mathbf{O}^{x}_{::p}\mathbf{F}_{::k'}$

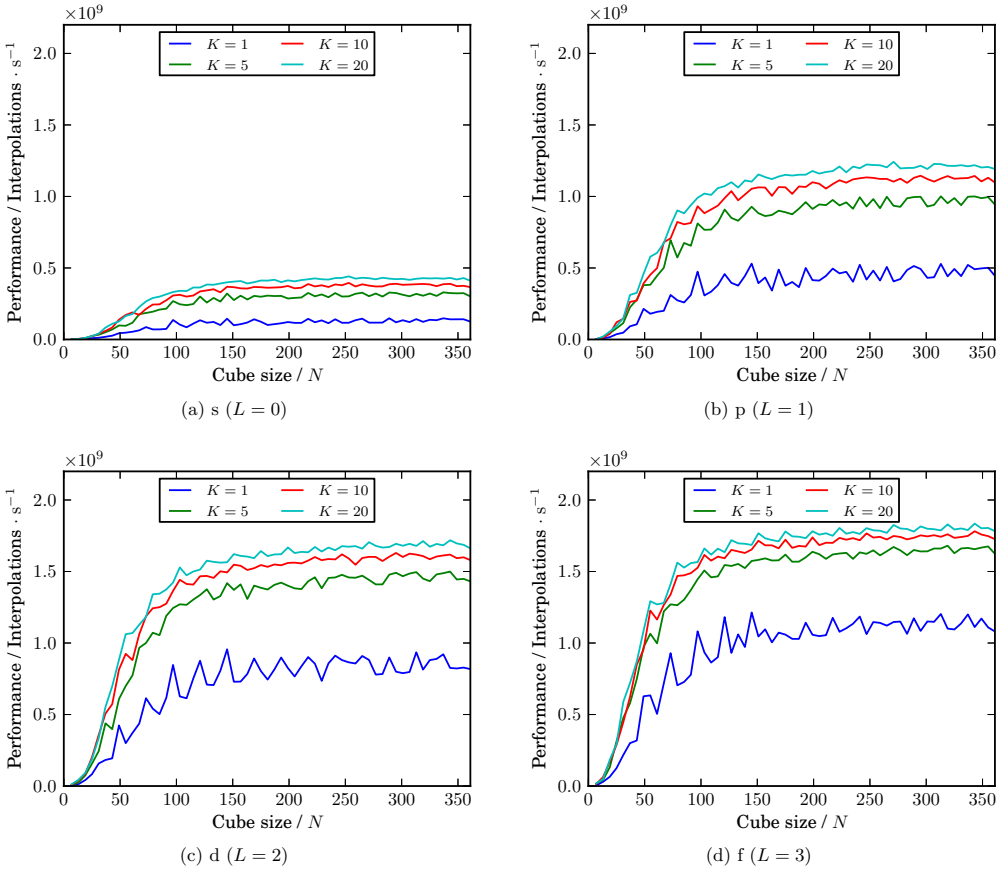
    $\mathbf{H}_{::k'} \leftarrow \mathbf{G}(\mathbf{O}^{y}_{::p})^{\mathbf{T}}$

  for all j slices along the y axis **do**

    $\mathbf{V}_{:j:} \leftarrow \mathbf{V}_{:j:} + \omega_p \mathbf{H}_{:j:}(\mathbf{O}^{z}_{::p})^{\mathbf{T}}$

$\mathbf{V} \leftarrow \mathbf{V} + C\mathbf{F}$

FIG. 5: Algorithm for the linear transformation.



(a) s ($L = 0$)

(b) p ($L = 1$)

(c) d ($L = 2$)

(d) f ($L = 3$)

FIG. 6: Performance for the bubbles injection in number of interpolations per second, for different values of L and number of bubbles K as a function of the number of points per dimension N.

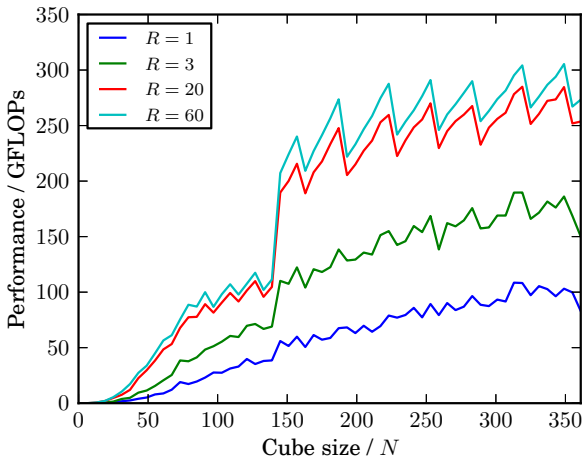


FIG. 7: Performance in GFLOPs for linear transformations with different operator rank R, as a function of the number of points per dimension N. The theoretical peak performance for a NVIDIA Tesla M2070 is 515 GFLOPs. The jump in performance at about $N = 150$ is due to cuBLAS switching to a different algorithm. Examples of transformations include identity/projection ($R = 1$), ∇^2 ($R = 3$), and convolution with the Coulomb kernel ($R = 20$ in the approximation used here).

TABLES

TABLE I: Errors and costs of the total Coulomb energies for a test set of molecules. For each molecule, results are reported for grid steps of 0.4, 0.2, 0.1 and 0.05 a_0. The number of coefficients required to store the bubbles and the cube are given in the N_{bub} and N_{cub} columns. ΔE is the error in the total Coulomb energy. t_* and $t_\times$ are the costs of the convolution using 20 quadrature points and the product, respectively. In parenthesis, we report the approximate cost per operation, $t_*/(120N_{\text{cub}}^{4/3}) \cdot 10^{-12}$, and $t_\times/(17KN_{\text{cub}}) \cdot 10^{-9}$.

Molecule	K	N_{bub}	N_{cub}	$\Delta E/E_{\text{h}}$	t_* / s	$t_\times$ / s
CO	2	$16{\cdot}10^3$	$118{\cdot}10^3$	$4.2{\cdot}10^{-3}$	0.02 (28.9)	0.01 (2.5)
			$701{\cdot}10^3$	$4.7{\cdot}10^{-3}$	0.10 (13.4)	0.07 (2.9)
			$6{\cdot}10^6$	$-1.5{\cdot}10^{-4}$	0.55 (4.7)	0.32 (1.7)
			$41{\cdot}10^6$	$6.1{\cdot}10^{-6}$	6.49 (3.9)	2.24 (1.6)
H_2	2	$10{\cdot}10^3$	$118{\cdot}10^3$	$3.0{\cdot}10^{-4}$	0.03 (43.4)	0.02 (5.0)
			$701{\cdot}10^3$	$-4.6{\cdot}10^{-5}$	0.09 (12.0)	0.06 (2.5)
			$5{\cdot}10^6$	$-1.2{\cdot}10^{-5}$	0.48 (4.5)	0.30 (1.7)
			$37{\cdot}10^6$	$-1.8{\cdot}10^{-5}$	5.58 (3.8)	2.07 (1.7)
CO_2	3	$24{\cdot}10^3$	$146{\cdot}10^3$	$-7.4{\cdot}10^{-2}$	0.03 (32.4)	0.03 (4.0)
			$788{\cdot}10^3$	$-3.3{\cdot}10^{-2}$	0.10 (11.5)	0.07 (1.7)
			$6{\cdot}10^6$	$-4.1{\cdot}10^{-4}$	0.61 (4.5)	0.43 (1.4)
			$44{\cdot}10^6$	$-4.4{\cdot}10^{-5}$	7.26 (3.8)	3.08 (1.4)
H_2O	3	$18{\cdot}10^3$	$118{\cdot}10^3$	$-4.5{\cdot}10^{-4}$	0.03 (43.4)	0.02 (3.3)
			$800{\cdot}10^3$	$-2.2{\cdot}10^{-4}$	0.11 (12.3)	0.07 (1.7)
			$6{\cdot}10^6$	$-1.4{\cdot}10^{-4}$	0.58 (4.5)	0.41 (1.4)
			$44{\cdot}10^6$	$-1.1{\cdot}10^{-6}$	7.06 (3.8)	2.96 (1.3)
O_3	3	$24{\cdot}10^3$	$146{\cdot}10^3$	$-5.3{\cdot}10^{-1}$	0.02 (21.6)	0.02 (2.7)
			$899{\cdot}10^3$	$5.0{\cdot}10^{-2}$	0.11 (10.6)	0.08 (1.7)
			$6{\cdot}10^6$	$2.4{\cdot}10^{-4}$	0.62 (4.5)	0.44 (1.4)
			$46{\cdot}10^6$	$8.6{\cdot}10^{-5}$	7.42 (3.7)	3.46 (1.5)
ZnH_2	3	$21{\cdot}10^3$	$146{\cdot}10^3$	$3.9{\cdot}10^{-2}$	0.02 (21.6)	0.02 (2.7)
			$874{\cdot}10^3$	$-2.6{\cdot}10^{-1}$	0.12 (12.0)	0.08 (1.8)
			$7{\cdot}10^6$	$-1.2{\cdot}10^{-2}$	0.67 (4.6)	0.46 (1.4)
			$47{\cdot}10^6$	$3.8{\cdot}10^{-4}$	7.67 (3.8)	3.25 (1.4)
NH_3	4	$22{\cdot}10^3$	$118{\cdot}10^3$	$1.1{\cdot}10^{-1}$	0.02 (28.9)	0.02 (2.5)
			$800{\cdot}10^3$	$-8.4{\cdot}10^{-4}$	0.10 (11.2)	0.08 (1.5)
			$6{\cdot}10^6$	$1.9{\cdot}10^{-4}$	0.66 (4.7)	0.52 (1.2)
			$50{\cdot}10^6$	$-2.7{\cdot}10^{-6}$	8.48 (3.8)	3.97 (1.2)
H_2O_2	4	$26{\cdot}10^3$	$146{\cdot}10^3$	1.4	0.03 (32.4)	0.03 (3.0)
			$899{\cdot}10^3$	$-1.4{\cdot}10^{-1}$	0.11 (10.6)	0.09 (1.5)
			$6{\cdot}10^6$	$4.2{\cdot}10^{-3}$	0.60 (4.3)	0.53 (1.2)
			$50{\cdot}10^6$	$1.5{\cdot}10^{-4}$	7.84 (3.6)	3.99 (1.2)

Molecule	K	N_{bub}	N_{cub}	$\Delta E/E_{\mathrm{h}}$	t_* / s	$t_\times$ / s
HCOOH	5	$33{\cdot}10^3$	$182{\cdot}10^3$	$-1.7{\cdot}10^{-1}$	0.02 (16.1)	0.04 (2.6)
			$1{\cdot}10^6$	$9.6{\cdot}10^{-3}$	0.15 (10.3)	0.14 (1.4)
			$8{\cdot}10^6$	$2.1{\cdot}10^{-3}$	0.80 (4.1)	0.78 (1.1)
			$62{\cdot}10^6$	$-1.4{\cdot}10^{-5}$	10.62 (3.6)	5.72 (1.1)
CH_4	5	$27{\cdot}10^3$	$118{\cdot}10^3$	$9.3{\cdot}10^{-3}$	0.02 (28.9)	0.03 (3.0)
			$913{\cdot}10^3$	$-3.0{\cdot}10^{-5}$	0.13 (12.2)	0.13 (1.7)
			$7{\cdot}10^6$	$-4.6{\cdot}10^{-5}$	0.78 (4.7)	0.72 (1.2)
			$52{\cdot}10^6$	$-6.3{\cdot}10^{-5}$	8.22 (3.5)	4.99 (1.1)
CH_3OH	6	$35{\cdot}10^3$	$227{\cdot}10^3$	$-3.8{\cdot}10^{-2}$	0.03 (18.1)	0.05 (2.2)
			$1{\cdot}10^6$	$-3.5{\cdot}10^{-2}$	0.17 (10.0)	0.18 (1.4)
			$9{\cdot}10^6$	$8.3{\cdot}10^{-4}$	0.90 (4.2)	0.93 (1.1)
			$64{\cdot}10^6$	$1.8{\cdot}10^{-4}$	10.94 (3.5)	6.62 (1.0)
H_2SO_4	7	$51{\cdot}10^3$	$227{\cdot}10^3$	$-1.1{\cdot}10^{-1}$	0.03 (18.1)	0.06 (2.2)
			$2{\cdot}10^6$	$2.9{\cdot}10^{-2}$	0.22 (9.8)	0.22 (1.2)
			$11{\cdot}10^6$	$-5.9{\cdot}10^{-3}$	1.31 (4.3)	1.37 (1.0)
			$88{\cdot}10^6$	$9.7{\cdot}10^{-4}$	16.08 (3.4)	10.00 (1.0)
CH_3COOH	8	$50{\cdot}10^3$	$182{\cdot}10^3$	$-7.9{\cdot}10^{-2}$	0.03 (24.2)	0.06 (2.4)
			$1{\cdot}10^6$	$-1.6{\cdot}10^{-2}$	0.18 (10.8)	0.21 (1.2)
			$10{\cdot}10^6$	$1.1{\cdot}10^{-3}$	1.07 (4.4)	1.27 (1.0)
			$72{\cdot}10^6$	$4.0{\cdot}10^{-4}$	13.29 (3.7)	9.12 (0.9)
C_2H_6	8	$44{\cdot}10^3$	$227{\cdot}10^3$	$-1.4{\cdot}10^{-2}$	0.03 (18.1)	0.07 (2.3)
			$1{\cdot}10^6$	$-8.9{\cdot}10^{-4}$	0.16 (9.4)	0.22 (1.2)
			$9{\cdot}10^6$	$4.7{\cdot}10^{-5}$	0.92 (4.3)	1.17 (1.0)
			$66{\cdot}10^6$	$-6.2{\cdot}10^{-5}$	11.47 (3.6)	8.61 (1.0)
C_2H_5OH	9	$52{\cdot}10^3$	$227{\cdot}10^3$	$-6.0{\cdot}10^{-2}$	0.03 (18.1)	0.07 (2.0)
			$1{\cdot}10^6$	$-4.1{\cdot}10^{-2}$	0.19 (9.8)	0.26 (1.2)
			$10{\cdot}10^6$	$-1.4{\cdot}10^{-3}$	1.07 (4.0)	1.47 (0.9)
			$74{\cdot}10^6$	$1.6{\cdot}10^{-4}$	13.44 (3.6)	10.40 (0.9)
C_3H_8	11	$61{\cdot}10^3$	$272{\cdot}10^3$	$-6.0{\cdot}10^{-2}$	0.04 (18.9)	0.09 (1.8)
			$2{\cdot}10^6$	$2.7{\cdot}10^{-3}$	0.22 (10.0)	0.32 (1.1)
			$11{\cdot}10^6$	$1.9{\cdot}10^{-5}$	1.11 (4.0)	1.79 (0.9)
			$80{\cdot}10^6$	$3.9{\cdot}10^{-5}$	14.64 (3.5)	12.90 (0.9)
C_6H_6	12	$74{\cdot}10^3$	$261{\cdot}10^3$	$3.5{\cdot}10^{-2}$	0.04 (20.0)	0.10 (1.9)
			$2{\cdot}10^6$	$1.4{\cdot}10^{-2}$	0.22 (10.6)	0.33 (1.1)
			$11{\cdot}10^6$	$-2.9{\cdot}10^{-4}$	1.22 (4.0)	2.03 (0.9)
			$83{\cdot}10^6$	$-3.6{\cdot}10^{-4}$	15.73 (3.6)	14.23 (0.8)
C_6F_6	12	$95{\cdot}10^3$	$261{\cdot}10^3$	-2.5	0.03 (15.0)	0.10 (1.9)
			$2{\cdot}10^6$	$-2.3{\cdot}10^{-1}$	0.24 (10.3)	0.34 (1.0)
			$12{\cdot}10^6$	$-1.4{\cdot}10^{-5}$	1.30 (4.0)	2.10 (0.9)
			$89{\cdot}10^6$	$-1.5{\cdot}10^{-3}$	16.51 (3.5)	15.22 (0.8)
Adenine	15	$101{\cdot}10^3$	$261{\cdot}10^3$	$1.2{\cdot}10^{-1}$	0.03 (15.0)	0.13 (2.0)
			$2{\cdot}10^6$	$9.9{\cdot}10^{-3}$	0.29 (10.5)	0.46 (1.0)
			$14{\cdot}10^6$	$-2.4{\cdot}10^{-4}$	1.51 (3.9)	2.87 (0.8)
			$105{\cdot}10^6$	$-3.5{\cdot}10^{-4}$	20.35 (3.4)	21.36 (0.8)